89/924,100

Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** FRO **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search | Clear | Interrupt

Search History

DATE: Wednesday, December 19, 2007 Purge Queries | Printable Copy | Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=OR			
L13	'6016484'.pn.	1	L13
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L12	l8 and l11	74	L12
L11	705/39	2473	L11
L10	6763336.pn.	2	L10
L9	L8 and bluetooth	16	L9
L8	L7 and (wireless or wirelessly) and (transmit or transmission)	459	L8
L7	L6 not @py>2001	529	L7
L6	L5 and 705.clas.	7707	L6
L5	L4 and (payment or invoice) and (information or data)	20718	L5
L4	l2 and L3	130500	L4
L3	(request or requestor) and (receive or reciever or recievor user or merchant)	510448	L3
L2	(wireless or bluetooth or "personal digital assistant" or "pda")and (transmit or transmission)	337857	L2

L1	(wireless or bluetooth or "personal digital assistant" or "pda")	594766	L1

END OF SEARCH HISTORY

Product search results for bluetooth>2000

Bluetooth Illustration	$1.00 - FeaturePics.com
Garmin Streetpilot 2820 Gps Navigation ...	$637.05 - ElectronicaDirect.com
Bluetooth Demystified by Nathan J. Muller ...	$20.43 - Dualj.com

See **bluetooth>2000** results available through Google Checkout

eBay Seller: wingwingdragon: Cell Phones, DUAL SIM TRIBAND ...

BLUETOOTH, **2000** GAMES, UNLOCKED, 3D SOUND, SUB-WOOFER. This seller accepts PayPal, Buy It Now or Best Offer, $189.99, $29.99, 22h 38m ...
cell-phones.search.ebay.com/_Cell-Phones-PDAs_W0QQsacatZ15032QQsassZwingwingdragon - 64k - Cached - Similar pages

Bluetooth News Archive 2000

At the **Bluetooth 2000** Congress last week in Monte Carlo, the company successfully sent files from a Sony laptop to a Hewlett-Packard LaserJet without any ...
www.palowireless.com/**Bluetooth**/newsarchive**2000**.asp - Similar pages

Coexistence Mechanisms for Interference Mitigation between IEEE ...

5 CCK Solutions to Avoid Mutual Interference (context) - Kamerman, **Bluetooth** - **2000** 5 Spread Spectrum Schemes for Microwave Frequency WLANs (context) ...
citeseer.ist.psu.edu/510605.html - 23k - Cached - Similar pages

[PDF] For Immediate Release

File Format: PDF/Adobe Acrobat - View as HTML
Both products were demonstrated at the IBC **Bluetooth 2000** conference in Monte Carlo. today. The companies will focus on environments such as conference ...
www.compactflash.org/pr/pr056.pdf - Similar pages

ONLINE, May 2001 | Bluetooth Bites Information Retrieval

According to Cahners In-Stat Group (http://www.instat.com) in its July **2000** report, "**Bluetooth 2000**: To Enable the Star Trek Generation," the manufacture of ...
www.onlinemag.net/OL2001/allen5_01.html - 20k - Cached - Similar pages

Bluetooth Wireless Chips Catapult to 1.4 Billion Units by 2005 ...

The report **Bluetooth 2000**: To Enable the Star Trek Generation (MM0009BW) forecasts **Bluetooth**-enabled equipment and the associated opportunities for radio ...
findarticles.com/p/articles/mi_m0EIN/is_**2000**_July_26/ai_63683449 - 25k - Cached - Similar pages

Wi-Fi and Bluetooth: Enabling Cooexistence

Bluetooth 2000: To Enable the Star Trek Generation, Report #MM00-098W (Scottsdale, AZ: Cahners In-Stat, July **2000**). 2. Enterprise Wireless LAN Market ...
www.ce-mag.com/archive/01/05/lansford.html - 47k - Cached - Similar pages

O'Reilly Network -- Personal Area Network: A **Bluetooth** Primer
Today, the **Bluetooth** SIG includes nearly **2000** companies, and prototype devices are beginning to make their way into the marketplace. ...
www.oreillynet.com/pub/a/wireless/**2000**/11/03/**bluetooth**.html - 24k - Cached - Similar pages

The Lastest **BlueTooth** Car Kits [Archive] - TreoCentral.com
Tim then replied without tipping his hand too much, "call me in January and ask me about **BlueTooth 2000**." By his response, I am concluding that this is a ...
discuss.treocentral.com/archive/index.php/t-65461.html - 150k - Cached - Similar pages

[PDF] Setting up a **Bluetooth** Packet Transport Link
File Format: PDF/Adobe Acrobat - View as HTML
http://www.cs.tut.fi/kurssit/83090/S13.ps. Referred. 5/10/**2000**. 4. Oraskari, Jyrki, **Bluetooth 2000**. http://www.hut.fi/joraskur/**bluetooth**.html ...
www.cs.hut.fi/~ctl/btpacket.pdf - Similar pages

News archive results for **bluetooth>2000**

2000 » Intel Pushes **Bluetooth** - PC World
2000 » Interoperability is key - $20.00 - Electronics Times
2000 » Event Summary Inkjet a Major Focus at PC Expo **2000** - $795.00 - GARTNER GROUP'S DATAQUEST

1 2 3 4 5 6 7 8 9 10 **Next**

bluetooth>2000 Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Term used: **bluetooth** Found **1,669** of **216,199**

Sort results by: relevance | Save results to a Binder | Try an Advanced Search

Display results: expanded form | Search Tips | Try this search in The ACM Guide

Open results in a new window

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Bluetooth: vision, goals, and architecture

Jaap Haartsen, Mahmoud Naghshineh, Jon Inouye, Olaf J. Joeressen, Warren Allen

October 1998 **ACM SIGMOBILE Mobile Computing and Communications Review**, Volume 2 Issue 4

Publisher: ACM

Full text available: pdf(953.51 KB) Additional Information: full citation, abstract, references

A few years ago it was recognized that the vision of a truly low-cost, low-power radio-based cable replacement was feasible. Such a ubiquitous link would provide the basis for portable devices to communicate together in an ad hoc fashion by creating *personal area networks* which have similar advantages to their office environment counterpart - the local area network (LAN). Bluetooth is an effort by a consortium of companies to design a royalty free technology specification enabling this ...

2 Shake 'em, but don't crack 'em: Cracking the Bluetooth PIN

Yaniv Shaked, Avishai Wool

June 2005 **Proceedings of the 3rd international conference on Mobile systems, applications, and services MobiSys '05**

Publisher: ACM Press

Full text available: pdf(223.67 KB) Additional Information: full citation, abstract, references, cited by, index terms

This paper describes the implementation of an attack on the Bluetooth security mechanism. Specifically, we describe a passive attack, in which an attacker can find the PIN used during the pairing process. We then describe the cracking speed we can achieve through three optimizations methods. Our fastest optimization employs an algebraic representation of a central cryptographic primitive (SAFER+) used in Bluetooth. Our results show that a 4-digit PIN can be cracked in less than 0.3 sec on an old ...

3 Mobile applications: Bluetooth and WAP push based location-aware mobile advertising system

Lauri Aalto, Nicklas Göthlin, Jani Korhonen, Timo Ojala

June 2004 **Proceedings of the 2nd international conference on Mobile systems, applications, and services MobiSys '04**

Publisher: ACM Press

Full text available: pdf(469.83 KB) Additional Information: full citation, abstract, references, citings, index terms

Advertising on mobile devices has large potential due to the very personal and intimate

nature of the devices and high targeting possibilities. We introduce a novel B-MAD system for delivering permission-based location-aware mobile advertisements to mobile phones using Bluetooth positioning and Wireless Application Protocol (WAP) Push. We present a thorough quantitative evaluation of the system in a laboratory environment and qualitative user evaluation in form of a field trial in the real envir ...

Keywords: Bluetooth positioning, context-aware, location-aware, location-based services, mobile advertising, wireless advertising

4 Quality-of-service in IP services over Bluetooth ad-hoc networks

Wah-Chun Chan, Jiann-Liang Chen, Po-Tsang Lin, Ka-Chin Yen

December 2003 **Mobile Networks and Applications**, Volume 8 Issue 6

Publisher: Kluwer Academic Publishers

Full text available: pdf(469.36 KB) Additional Information: full citation, abstract, references, index terms

Along with the development of multimedia and wireless networking technologies, mobile multimedia applications are playing more important roles in information access. Quality of Service (*QoS*) is a critical issue in providing guaranteed service in a low bandwidth wireless environment. To provide Bluetooth-IP services with differentiated quality requirements, a *QoS*-centric cascading mechanism is proposed in this paper. This innovative mechanism, composed of intra-piconet resource allo ...

Keywords: BNEP protocol, Bluetooth-IP access system, handoff, quality of service, resource allocation

5 Performance of Bluetooth bridges in scatternets with limited service scheduling

Vojislav B. Mišic´, Jelena Mišic´

February 2004 **Mobile Networks and Applications**, Volume 9 Issue 1

Publisher: Kluwer Academic Publishers

Full text available: pdf(552.34 KB) Additional Information: full citation, abstract, references, index terms, review

The performance of two Bluetooth piconets linked through a bridge device is analyzed using the tools of queueing theory. We analyze both possible cases, i.e., when the bridge device is the master in one of the piconets and a slave in the other (MS bridge), as well as when the bridge device is the slave in both of the piconets (SS bridge). Analytical results are derived for the probability distribution of access delay (i.e., the time that a packet has to wait before being serviced) and end-to-end ...

Keywords: Bluetooth, Bluetooth scatternet, master/slave bridge, performance evaluation, queueing theory, slave/slave bridge

6 Bluetooth dynamic scheduling and interference mitigation

N. Golmie

February 2004 **Mobile Networks and Applications**, Volume 9 Issue 1

Publisher: Kluwer Academic Publishers

Full text available: pdf(194.60 KB) Additional Information: full citation, abstract, references, index terms, review

Bluetooth is a cable replacement technology for Wireless Personal Area Networks. It is designed to support a wide variety of applications such as voice, streamed audio and video, web browsing, printing, and file sharing, each imposing a number of quality of service constraints including packet loss, latency, delay variation, and throughput. In addition to QOS support, another challenge for Bluetooth stems from having to share the

2.4 GHz ISM band with other wireless devices such as IEEE 802.11. ...

Keywords: Bluetooth, MAC scheduling, WPANs, interference

7 Platforms: Bluetooth and sensor networks: a reality check

Martin Leopold, Mads Bondo Dydensborg, Philippe Bonnet
November 2003 **Proceedings of the 1st international conference on Embedded networked sensor systems SenSys '03**
Publisher: ACM Press

Full text available: pdf(356.11 KB) Additional Information: full citation, abstract, references, citings, index terms

The current generation of sensor nodes rely on commodity components. The choice of the radio is particularly important as it impacts not only energy consumption but also software design (e.g., network self-assembly, multihop routing and in-network processing). Bluetooth is one of the most popular commodity radios for wireless devices. As a representative of the frequency hopping spread spectrum radios, it is a natural alternative to broadcast radios in the context of sensor networks. The questio ...

Keywords: bluetooth, mac layer, network self-assembly, sensor nodes

8 Rendezvous layer protocols for Bluetooth-enabled smart devices

Frank Siegemund, Michael Rohs
July 2003 **Personal and Ubiquitous Computing**, Volume 7 Issue 2
Publisher: Springer-Verlag
Full text available: pdf(445.27 KB) Additional Information: full citation, abstract, index terms

AbstractCommunication platforms for ubiquitous computing need to be flexible, self-organizing, highly scalable and energy efficient, because in the envisioned scenarios a large number of autonomous entities communicate in potentially unpredictable ways. Short-range wireless technologies form the basis of such communication platforms. In this paper we investigate device discovery in Bluetooth, a candidate wireless technology for ubiquitous computing. Detecting new devices accounts for a significa ...

Keywords: Adaptive rendezvous protocols, Bluetooth, Context, Cooperative device discovery, Energy efficiency, Inquiry parameters, Rendezvous layer

9 Emerging technologies: WLANs and WPANs: On the application of traffic engineering over bluetooth ad hoc networks

Sachin Abhyankar, Rishi Toshiwal, Carlos Cordeiro, Dharma Agrawal
September 2003 **Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems MSWIM '03**
Publisher: ACM Press
Full text available: pdf(420.12 KB) Additional Information: full citation, abstract, references, index terms

The seamless communication of data and voice over short-range, point-to-multipoint wireless links between mobile and/or stationary devices is becoming a reality by newly introduced Bluetooth radio technology for Wireless Personal Area Networking, which can support only up to 1 Mbps of nominal bandwidth. It is based on a master-slave model where double the resources are allocated for any slave-to-slave communication via the master. In addition, it does not have any mechanism to serve demands exce ...

Keywords: bluetooth, performance evaluation, piconet partitioning, role switching, traffic engineering

10 Bluetooth: a technical overview

Myra Dideles
June 2003 **Crossroads**, Volume 9 Issue 4
Publisher: ACM Press
Full text available: htm(38.37 KB) Additional Information: full citation, references, index terms

11 An ns-based Bluetooth Topology Construction Simulation Environment

Chia-Jui Hsu, Yuh-Jzer Joung
March 2003 **Proceedings of the 36th annual symposium on Simulation ANSS '03**
Publisher: IEEE Computer Society
Full text available: pdf(470.02 KB) Additional Information: full citation, abstract, index terms

Bluetooth is an emerging technology in wireless applications,and many related issues are yet to be explored bothin academia and industry. Because of the complexity andthe dynamics of computer networks, a good simulation toolplays an important role in the development stage. Of the existingsimulation tools, ns is a popular, open-source packagethat has a substantial support for simulation of TCP,routing, and multicast protocols over wired and wirelessnetworks. It also has BlueHoc as its extension f ...

12 Interference evaluation of Bluetooth and IEEE 802.11b systems

N. Golmie, R. E. Van Dyck, A. Soltanian, A. Tonnerre, O. Rébala
May 2003 **Wireless Networks**, Volume 9 Issue 3
Publisher: Kluwer Academic Publishers
Full text available: pdf(203.73 KB) Additional Information: full citation, abstract, references, citings, index terms

The emergence of several radio technologies, such as Bluetooth and IEEE 802.11, operating in the 2.4 GHz unlicensed ISM frequency band, may lead to signal interference and result in significant performance degradation when devices are colocated in the same environment. The main goal of this paper is to evaluate the effect of mutual interference on the performance of Bluetooth and IEEE 802.11b systems. We develop a simulation framework for modeling interference based on detailed MAC and PHY model ...

Keywords: Bluetooth, IEEE 802.11b, WPANs, interference

13 Bluetooth: A pseudo random coordinated scheduling algorithm for Bluetooth scatternets

András Rácz, György Miklós, Ferenc Kubinszky, Andrés Valkó
October 2001 **Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing MobiHoc '01**
Publisher: ACM Press
Full text available: pdf(218.47 KB) Additional Information: full citation, abstract, references, citings, index terms

The emergence of Bluetooth as a default radio interface allows handheld devices to be rapidly interconnected into ad hoc networks. Bluetooth allows large numbers of piconets to form a scatternet using designated nodes that participate in multiple piconets. A unit that participates in multiple piconets can serve as a bridge and forwards traffic between neighbouring piconets. Since a Bluetooth unit can transmit or receive in only one piconet at a time, a bridging unit has to share its time among t ...

Keywords: Bluetooth, inter-piconet communication, scatternet, scheduling

14 Multimedia support over bluetooth Piconets

Rohit Kapoor, Manthos Kazantzidis, Mario Gerla, Per Johansson
July 2001 **Proceedings of the first workshop on Wireless mobile internet WMI '01**
Publisher: ACM Press

Full text available: pdf(808.15 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we explore the ability to support multimedia traffic in indoor, wireless ad hoc PANs (Personal Area Networks) using the Bluetooth technology. We first define the representative ad hoc networking applications such as wireless access to the Internet, document distribution, videoconferencing, webcasting, interaction with sensors and actuators, etc. For such applications, we define the performance requirements placed on the PAN. There are two technologies now competing for the ...

Keywords: Piconets, WaveLAN, bluetooth, multimedia, video, voice

15 Interference of bluetooth and IEEE 802.11: simulation modeling and performance evaluation

N. Golmie, R. E. Van Dyck, A. Soltanian
July 2001 **Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems MSWIM '01**
Publisher: ACM Press

Full text available: pdf(657.91 KB) Additional Information: full citation, abstract, references, citings, index terms

The emergence of several radio technologies such as Bluetooth, and IEEE 802.11 operating in the 2.4 GHz unlicensed ISM frequency band may lead to signal interference and result in significant performance degradation when devices are co-located in the same environment. The main goal of this paper is to present a simulation environment for modeling interference based on detailed MAC and PHY models. This framework is then used to evaluate the impact of interference on the performance of Bluetoot ...

Keywords: IEEE 802.11, WPANs, bluetooth, interference

16 Emerging threats: A preliminary investigation of worm infections in a bluetooth environment

Jing Su, Kelvin K. W. Chan, Andrew G. Miklas, Kenneth Po, Ali Akhavan, Stefan Saroiu, Eyal de Lara, Ashvin Goel
November 2006 **Proceedings of the 4th ACM workshop on Recurring malcode WORM '06**
Publisher: ACM Press

Full text available: pdf(876.85 KB) Additional Information: full citation, abstract, references, index terms

Over the past year, there have been several reports of malicious code exploiting vulnerabilities in the Bluetooth protocol. While the research community has started to investigate a diverse set of Bluetooth security issues, little is known about the feasibility and the propagation dynamics of a worm in a Bluetooth environment. This paper is an initial attempt to remedy this situation.We start by showing that the Bluetooth protocol design and implementation is large and complex. We gather traces ...

Keywords: Bluetooth, malware, worms

17 Performance monitoring: Performance evaluation of web services invocation over Bluetooth

Vincenzo Auletta, Carlo Blundo, Emiliano De Cristofaro, Guerriero Raimato

October 2006 **Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks PM2HW2N '06**

Publisher: ACM Press

Full text available: pdf(506.38 KB) Additional Information: full citation, abstract, references, index terms

Mobile devices should allow users to exploit services anytime, without any place restriction and in a transparent way. The Bluetooth technology achieves this feature, by providing services at a low cost and with a low power consumption. In a few years, most of the devices accessing the web services will be mobile. In this paper, we evaluate performance of a framework allowing a Java application programmer to directly interface Web Services from a mobile device using a Bluetooth connection. Our w ...

18 Quality of service: Evaluation of the energetic impact of Bluetooth low-power modes for ubiquitous computing applications

Juan-Carlos Cano, José-Manuel Cano, Eva González, Carlos Calafate, Pietro Manzoni

October 2006 **Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks PE-WASUN '06**

Publisher: ACM Press

Full text available: pdf(756.49 KB) Additional Information: full citation, abstract, references, index terms

In order to further increase the applicability of Bluetooth in real applications, reducing the energy consumption and hardware cost are important research topics. In this paper we present a wireless communication prototype to support ubiquitous computing, which has been implemented based on commercial Bluetooth off-the-shelf components. It allows every object to be augmented with processing and communication capabilities in order to make them "smart". We investigate on the power characteristics ...

Keywords: Bluetooth, Bluetooth measurements, power consumption, ubiquitous computing

19 Service support: SymPhone: design and implementation of a VoIP peer for Symbian mobile phones using Bluetooth and SIP

Patrick Stuedi, Andreas Frei, Luc Burdet, Gustavo Alonso

September 2006 **Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots WMASH '06**

Publisher: ACM Press

Full text available: pdf(454.17 KB) Additional Information: full citation, abstract, references, index terms

VoIP is born from the growing Internet infrastructure, which has over the years seen significant improvements in both bandwidth and end-to-end latency. In this paper, we explore making VoIP available on a mobile phone. For that purpose, we propose an architecture and describe the various components involved. Data entering and leaving the mobile phone is encapsulated in a wireless Bluetooth connection. The bridge to the Internet is provided by a linux Bluetooth access point. The system is compati ...

Keywords: SIP, VoIP, bluetooth, symbian

20 Papers from MC²R open call: Using visual tags to bypass Bluetooth device discovery

David Scott, Richard Sharp, Anil Madhavapeddy, Eben Upton

January 2005 **ACM SIGMOBILE Mobile Computing and Communications Review**, Volume

9 Issue 1

Publisher: ACM Press

Full text available: pdf(311.14 KB) Additional Information: full citation, abstract, references, citings, index terms

One factor that has limited the use of Bluetooth as a networking technology for publicly accessible mobile services is the way in which it handles Device Discovery. Establishing a Bluetooth connection between two devices that have not seen each other before is slow and, from a usability perspective, often awkward. In this paper we present the implementation of an end-to-end Bluetooth-based mobile service framework designed specifically to address this issue. Rather than using the standard Blueto ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

☐ Search Results

Results for "((bluetooth)<in>metadata)"
Your search matched **2092** of **1706580** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

Modify Search

((bluetooth)<in>metadata) Search

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Search Options

View Session History

New Search

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

IEEE/IET Books Educational Courses A

IEEE/IET journals, transactions, letters, magazines, conference proceedings, and

view selected items Select All Deselect All View: 1

☐ 1. **Performance evaluation of the Bluetooth-based public Internet access p**
Yujin Lim; Jesung Kim; Sang Lyul Min; Joong Soo Ma;
Information Networking, 2001. Proceedings. 15th International Conference on
31 Jan.-2 Feb. 2001 Page(s):643 - 648
Digital Object Identifier 10.1109/ICOIN.2001.905527
AbstractPlus | Full Text: PDF(540 KB) IEEE CNF
Rights and Permissions

☐ 2. **A novel architecture and coexistence method to provide global access t by IEEE 802.11 WLANs**
Cordeiro, C.D.M.; Abhyankar, S.; Toshiwal, R.; Agrawal, D.P.;
Performance, Computing, and Communications Conference, 2003. Conferenc
IEEE International
9-11 April 2003 Page(s):23 - 30
AbstractPlus | Full Text: PDF(906 KB) IEEE CNF
Rights and Permissions

☐ 3. **Simplifications of the Bluetooth radio devices**
Youquan Zheng; Zhenming Feng;
Networked Appliances, 2002. Gaithersburg. Proceedings. 2002 IEEE 4th Inte
2002 Page(s):107 - 115
Digital Object Identifier 10.1109/IWNA.2001.980816
AbstractPlus | Full Text: PDF(202 KB) IEEE CNF
Rights and Permissions

☐ 4. **What is Bluetooth?**
McDermott-Wells, P.;
Potentials, IEEE
Volume 23, Issue 5, Dec 2004-Jan 2005 Page(s):33 - 35
Digital Object Identifier 10.1109/MP.2005.1368913
AbstractPlus | Full Text: PDF(545 KB) IEEE JNL
Rights and Permissions

5. **Bluetooth Performance Analysis in Personal Area Network (PAN)**

Rashid, R.A.; Yusoff, R.;
RF and Microwave Conference, 2006. RFM 2006. International
12-14 Sept. 2006 Page(s):393 - 397
Digital Object Identifier 10.1109/RFM.2006.331112
AbstractPlus | Full Text: PDF(4436 KB) IEEE CNF
Rights and Permissions

6. **A key establishment protocol for Bluetooth scatternets**
Li, H.; Mukesh Singhal;
Distributed Computing Systems Workshops, 2005. 25th IEEE International Cc
6-10 June 2005 Page(s):610 - 616
Digital Object Identifier 10.1109/ICDCSW.2005.14
AbstractPlus | Full Text: PDF(95 KB) IEEE CNF
Rights and Permissions

7. **Residential microwave oven interference on Bluetooth data performanc**
Rondeau, T.W.; D'Souza, M.F.; Sweeney, D.G.;
Consumer Electronics, IEEE Transactions on
Volume 50, Issue 3, Aug. 2004 Page(s):856 - 863
Digital Object Identifier 10.1109/TCE.2004.1341691
AbstractPlus | Full Text: PDF(553 KB) IEEE JNL
Rights and Permissions

8. **Power-efficient and QoS-aware scheduling in Bluetooth scatternet for w**
Yang-Ick Joo; Tae-Jin Lee; Doo Seop Eom; Yeonwoo Lee; Kyun Hyon Tchah
Consumer Electronics, IEEE Transactions on
Volume 49, Issue 4, Nov. 2003 Page(s):1067 - 1072
Digital Object Identifier 10.1109/TCE.2003.1261197
AbstractPlus | Full Text: PDF(456 KB) IEEE JNL
Rights and Permissions

9. **Employing dynamic segmentation for effective co-located coexistence b IEEE 802.11 WLANs**
de M Cordeiro, C.; Agrawal, D.P.;
Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE
Volume 1, 17-21 Nov. 2002 Page(s):195 - 200 vol.1
AbstractPlus | Full Text: PDF(501 KB) IEEE CNF
Rights and Permissions

10. **Rapid heterogeneous ad hoc connection establishment: accelerating Bl IrDA**
Woodings, R.W.; Joos, D.D.; Clifton, T.; Knutson, C.D.;
Wireless Communications and Networking Conference, 2002. WCNC2002. 2(
Volume 1, 17-21 March 2002 Page(s):342 - 349 vol.1
Digital Object Identifier 10.1109/WCNC.2002.993519
AbstractPlus | Full Text: PDF(442 KB) IEEE CNF
Rights and Permissions

11. **Mutual interference between independent Bluetooth piconets**
Howitt, I.;
Vehicular Technology, IEEE Transactions on
Volume 52, Issue 3, May 2003 Page(s):708 - 718
Digital Object Identifier 10.1109/TVT.2003.811614
AbstractPlus | References | Full Text: PDF(845 KB) IEEE JNL
Rights and Permissions

12. **Near Field Communication and Bluetooth Bridge System for Mobile Con**
Leong, C.Y.; Ong, K.C.; Tan, K.K.; Gan, O.P.;
Industrial Informatics, 2006 IEEE International Conference on

Aug. 2006 Page(s):50 - 55
Digital Object Identifier 10.1109/INDIN.2006.275716

AbstractPlus | Full Text: PDF(358 KB) IEEE CNF
Rights and Permissions

13. **Bluetooth simulations for wireless sensor networks using GTNetS**
Xin Zhang; Riley, G.F.;
Modeling, Analysis, and Simulation of Computer and Telecommunications Sy 2004). Proceedings. The IEEE Computer Society's 12th Annual International
4-8 Oct. 2004 Page(s):375 - 382
Digital Object Identifier 10.1109/MASCOT.2004.1348292

AbstractPlus | Full Text: PDF(368 KB) IEEE CNF
Rights and Permissions

14. **A remotely controlled Bluetooth enabled environment**
Chakrabarti, S.; Liyun Wu; Son Vuong; Leung, V.C.M.;
Consumer Communications and Networking Conference, 2004. CCNC 2004.
5-8 Jan. 2004 Page(s):77 - 81
Digital Object Identifier 10.1109/CCNC.2004.1286836

AbstractPlus | Full Text: PDF(1458 KB) IEEE CNF
Rights and Permissions

15. **FPQ: a fair and efficient polling algorithm with QoS support for Bluetoot**
Lapeyrie, J.-B.; Turletti, T.;
INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Comp Societies. IEEE
Volume 2, 30 March-3 April 2003 Page(s):1322 - 1332 vol.2
Digital Object Identifier 10.1109/INFCOM.2003.1208968

AbstractPlus | Full Text: PDF(417 KB) IEEE CNF
Rights and Permissions

16. **Providing mobile LAN access capability for Bluetooth devices**
Shih-Yen Chiu; Hsung-Pin Chang; Ruei-Chuan Chang;
Parallel and Distributed Systems, 2002. Proceedings. Ninth International Con
17-20 Dec. 2002 Page(s):631 - 636
Digital Object Identifier 10.1109/ICPADS.2002.1183483

AbstractPlus | Full Text: PDF(308 KB) IEEE CNF
Rights and Permissions

17. **Basestation collaboration in Bluetooth voice networks**
Jingxin Xue; Todd, T.D.;
Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE
14-16 Nov. 2001 Page(s):533 - 538
Digital Object Identifier 10.1109/LCN.2001.990833

AbstractPlus | Full Text: PDF(528 KB) IEEE CNF
Rights and Permissions

18. **Integrating Bluetooth with wireless and ricocheting**
Lee, D.Y.J.; Lee, W.C.Y.;
Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The Symposium on
Volume 2, 18-21 Sept. 2000 Page(s):1310 - 1314 vol.2
Digital Object Identifier 10.1109/PIMRC.2000.881631

AbstractPlus | Full Text: PDF(304 KB) IEEE CNF
Rights and Permissions

19. **Scenario driven evaluation and interference mitigation proposals for Blu Bluetooth enabled consumer electronic devices**
Arumugam, A.K.; Nix, A.R.; Fletcher, P.N.; Armour, S.M.D.; Lee, B.S.;

Consumer Electronics, IEEE Transactions on
Volume 48, Issue 3, Aug. 2002 Page(s):754 - 764
Digital Object Identifier 10.1109/TCE.2002.1037071

AbstractPlus | Full Text: PDF(1358 KB) IEEE JNL
Rights and Permissions

20. **An Integrated Neuro-Fuzzy Approach to MPEG Video Transmission in B**
Kazemian, H.B.; Chantaraskul, S.;
Computational Intelligence in Image and Signal Processing, 2007. CIISP 200
1-5 April 2007 Page(s):287 - 292
Digital Object Identifier 10.1109/CIISP.2007.369183

AbstractPlus | Full Text: PDF(368 KB) IEEE CNF
Rights and Permissions

21. **A bypassing security model for anonymous Bluetooth peers**
Hahnsang Kim; Dabbous, W.; Afifi, H.;
Wireless Networks, Communications and Mobile Computing, 2005 Internatior
Volume 1, 13-16 June 2005 Page(s):310 - 315 vol.1
Digital Object Identifier 10.1109/WIRLES.2005.1549428

AbstractPlus | Full Text: PDF(1568 KB) IEEE CNF
Rights and Permissions

22. **Mitigating the effects of intermittent interference on Bluetooth ad hoc ne**
de M Cordeiro, C.; Agrawal, D.P.;
Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE In
Volume 1, 15-18 Sept. 2002 Page(s):496 - 500 vol.1

AbstractPlus | Full Text: PDF(426 KB) IEEE CNF
Rights and Permissions

23. **Rendezvous scheduling in Bluetooth scatternets**
Johansson, P.; Kapoor, R.; Kazantzidis, A.; Gerla, M.;
Communications, 2002. ICC 2002. IEEE International Conference on
Volume 1, 28 April-2 May 2002 Page(s):318 - 324
Digital Object Identifier 10.1109/ICC.2002.996868

AbstractPlus | Full Text: PDF(3072 KB) IEEE CNF
Rights and Permissions

24. **A routing vector method (RVM) for routing in Bluetooth scatternets**
Bhagwat, P.; Segall, A.;
Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE Internatic
15-17 Nov. 1999 Page(s):375 - 379
Digital Object Identifier 10.1109/MOMUC.1999.819514

AbstractPlus | Full Text: PDF(548 KB) IEEE CNF
Rights and Permissions

25. **A novel family of frequency hopping sequences for multi-hop Bluetooth**
Zan Li; Yilin Chang; Lijun Jin;
Consumer Electronics, IEEE Transactions on
Volume 49, Issue 4, Nov. 2003 Page(s):1084 - 1089
Digital Object Identifier 10.1109/TCE.2003.1261200

AbstractPlus | Full Text: PDF(490 KB) IEEE JNL
Rights and Permissions

View: 1

Help Contact Us

Please scan documents into application 09/924,100 Thanks.

Performance Evaluation of the Bluetooth-based Public Internet Access Point

Yujin Lim†, Jesung Kim, Sang Lyul Min, and Joong Soo Ma‡

Department of Mechanical Engineering, University of Seoul, Seoul, Korea†
School of Computer Science and Engineering, Seoul National University, Seoul, Korea
Information and Communications University, Taejon, Korea‡

yujin@uos.ac.kr jskim@archi.snu.ac.kr symin@dandelion.snu.ac.kr jsma@icu.ac.kr

Abstract— **Recently, Bluetooth has been regarded as a promising solution to an inexpensive wireless connection. Although initial application of Bluetooth technology has been focused mainly on replacing cables between hand-held devices, general wireless telecommunication such as public Internet access via a Bluetooth-equipped device is expected to be one of the most popular applications in the near future. However, it is not well understood whether performance of Bluetooth-based systems is sufficient for such an application. In this paper, we present preliminary results of performance evaluation of a Bluetooth-based Internet access point. The evaluation is based on simulation of an Internet access model consisting of a Bluetooth-based network access point and a number of Bluetooth-equipped notebook computers. The simulation results indicate that Bluetooth provides performance comparable to the fastest dial-up modem even when a number of users share a single Bluetooth radio unit. Better performance is expected when more than one Bluetooth radio unit are employed such that each unit services different users concurrently. However, Bluetooth units in a single radio range (about 10m) interferes each other since the channels established by each unit occasionally collide at the same frequency band. This paper analyzes the impact of such interference based on an analytical model of inter-channel interference. The analysis shows that the performance improves as the number of channels increases up to 40. We expect the proposed inter-channel interference model is useful in the design of systems facilitating multiple Bluetooth units.**

I. Introduction

Interconnection of consumer devices has been increasingly important as the use of hand-held devices such as PDA's and celluar phones becomes increasingly popular. In the past, these devices were usually connected via a serial cable that requires physical wiring between two devices before they can exchange information. To avoid such inconvenience, a wireless communication technology has highly been demanded by users. Bluetooth is one of such technologies and especially attractive in the environment of hand-held devices since a low cost is an important requirement of such a technology [1]. Due to its low cost, Bluetooth is expected to be embedded in many consumer devices in the near future. Combined with the demand for access to the Internet everywhere in public places, a Bluetooth-based public Internet access point is expected to be popular.

While protocol models for Bluetooth-based access networks have been proposed in the literature [2][3], their performance is yet to be known. In this paper, we present a simulation model of Internet access based on the Bluetooth technology. The model consists of an Internet access point and a number of user devices each equipped with a Bluetooth radio. To evaluate the performance of the model, we have developed a simulator that accurately models the behavior of each user's requests to Internet access and actions occurred in the Internet access point to service the requests. Since the radio in the Internet access point is shared by multiple users, a suitable scheduling policy is necessary to select which user will be serviced next. A plain round-robin policy seems to be a good choice since it is simple to implement yet provides fairness. We also consider two other scheduling policies called a *weighted round-robin* and a *multi-level round-robin*. The simulation results show that both policies improve the performance by up to 35% over the plain round-robin policy.

Performance can further be improved by facilitating multiple Bluetooth radios in the Internet access point. It would be ideal if the performance boosts up to N times when N Bluetooth radios are used. However, since more than one radio may use the same carrier band due to the nature of Frequency Hopping Spread Spectrum (FHSS) [4], the effective performance is less than N times especially when N becomes larger. This paper analyzes the performance of a Bluetooth system with multiple radios using a probabilistic model reflecting the interference.

This paper is organized as follows. The following section describes the Internet access model and the simulation methodology. Section III presents the simulation results along with detailed analysis. Section IV analyzes the impact of multiple Bluetooth radio units in the Internet access point. Finally, Section V gives concluding remarks.

II. Evaluation Model and Methodology

A Bluetooth-based Internet access point is a system that provides Bluetooth-equipped devices with access to the Internet. It consists of a Bluetooth radio unit through which data communications occur with user devices, and an interface to the wire-based network infrastructure leading to the Internet. The main role of the Internet access point is to establish a wireless connection to each user device and forward user's packets to/from the wired network.

A Bluetooth-based Internet access point may have either

This research was supported in part by the Ministry of Education under the BK21 program. The first author was supported from the Basic Research Program (for the woman scientists) of the Korea Science & Engineering Foundation.

0-7695-0951-7/01 $10.00 © 2001 IEEE

a single or multiple radio units. Also, the radio units may act as a master or a slave. This gives us a design space encompassing four possible combinations: (1) single slave unit, (2) single master unit, (3) multiple slave units, (4) multiple master units. In the first combination, the Internet access point acts as a slave while each user device acts as a master establishing it's own piconet to connect to the Internet access point. In this setup, the total bandwidth of the radio link is shared by all of the user devices. However, this design requires the slave unit to switch from one piconet to another, which takes up to one frame. This overhead is significant when switching occurs frequently, e.g., in the order of a few frames.

The second design choice is the Internet access point with a single radio unit that acts as a master of a piconet with all user devices as slaves. When the radio is in the master mode, it controls all the slaves participating in the piconet by determining which slave exchanges data with the master. This design is more efficient than the first design since the overhead of piconet switching is not present.

The remaining two designs are extensions of the two initial designs with additional radio units in the Internet access point. This allows load-sharing between radio units and hence improves both the total bandwidth and the maximum number of users, at the costs of additional radio units. However, multiple radio units may interfere each other by occasionally colliding at the same frequency band, whose performance impact is analyzed in Section IV.

In the simulation, we assume the second design. We also assume that the number of slaves is limited to seven in order to remove the effect of parking that is required when more than seven slaves are to be connected to a master. We focus on the throughput and delay per user in a steady-state environment where no users join or depart the piconet. For the evaluation, we have developed an event-driven, packet-level simulator using PARSEC developed at UCLA as a successor to Maisie [5].

As for the Internet access model, we use a closed queuing simulation model that processes Internet web surfing traffic as shown in Figure 1. In this model, a user with a notebook generates traffic by clicking on the screen after consuming a *UserThinkTime*. The user's request is inserted into a upload queue in which the transmission is delayed until the master's polling is received. This delay is called a *Bluetooth upload scheduling delay* and dependent on the scheduling policy of the master. After the *upload scheduling delay*, the notebook that is polled by the master sends the user's request to the Internet Access Point(AP) and the AP forwards the request to the Internet. *InternetAccessTime* is the time for the AP to receive the user-requested data from the Internet since the request is submitted. When the requested data is ready, the data is inserted into the download queue and stays there until the corresponding notebook is polled. This delay is called a *Bluetooth download scheduling delay* and also affected by the scheduling policy of the master. At this time, AP performs segmentation if necessary based on the length of the message.

The packets used in the experiments include NULL and

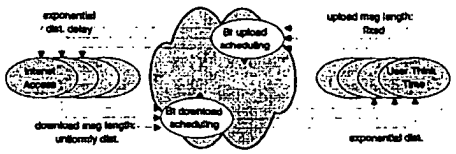


Fig. 1. Bluetooth simulation model

TABLE I
PARAMETER VALUES FOR EXPERIMENT

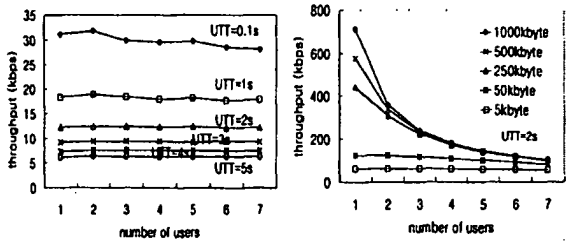
Parameter	Average Values
Message Length	[5 Kbytes, 50 Kbytes, 250 KBytes 500 Kbytes, 1000 Kbytes]
InternetAccessTime	[0.5s, 1s]
UserThinkTime	[0.1s, 1s, 2s, 3s, 4s, 5s]

POLL packets for control data transmission, and DH1, DH3, and DH5 packets for user data transmission. We assume fixed-length upload data and variable-length download data reflecting the asymmetric nature of the Internet traffic. The parameter values used in the simulation are shown in Table I. The *InternetAccessTime* and the *UserThinkTime* are random variables distributed exponentially with averages shown in the table. The message length is also a random variable, distributed uniformly. Each simulation is performed for a window of 24 hours.

We measure the performance by throughput and delay. The throughput is defined as the rate of data transmitted to a slave in a time unit. As for the delay, two measurements are considered: *completion delay* and *user delay*. The user delay is defined as the time between the user's click and the arrival of the first packet of the requested data. It is equal to the sum of *Bluetooth upload scheduling delay*, *InternetAccessTime*, and *download scheduling delay*. The completion delay is defined as the time between the user's click and the arrival of the last packet of the requested data. The completion delay represents the time until the requested service is completed.

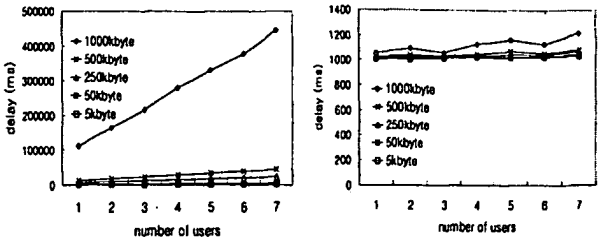
As mentioned before, the Bluetooth upload and download scheduling delays are affected by the scheduling policy of the master. In Bluetooth specification 1.0, a round-robin (RR) is implicitly assumed as the underlying scheduling policy. In an RR policy, a master polls its slaves one by one in a fixed order. A slave is allowed to transmit a message in a designated slot only when it has been addressed by the master in the preceding slot. Thus the polling sequence of the master is critical to the Bluetooth scheduling delay. The RR policy is fair in the sense that each user receives the same number of polls. Although it seems fair, a slave with download data should wait for other slaves even when the other slaves do not need to be serviced.

In this paper, we consider two other scheduling policies to improve the delay: *weighted round robin* (weighted RR) and *multi-level round robin* (multi-level RR). In the weighted RR, the master polls a slave up to n times successively when there is data that needs to be transmitted to the slave, while slaves with no data is polled only once.



(a) Message length = 5 Kbytes (b) Various message length

Fig. 2. Throughput per user



(a) Completion delay (b) User delay

Fig. 3. The variation of delay for the message length

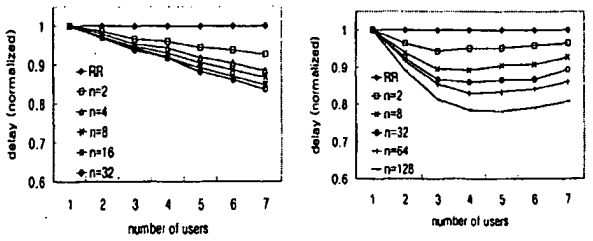
This policy improves the completion delay as we will see in the next section. On the other hand, the multi-level RR can improve the user delay, as well as the completion delay by grouping the slaves into two classes, one consisting of slaves with download data and the other consisting of slaves with no data. In this policy, the master polls all the slaves belonging to the first class up to n times successively, prior to polling the slaves belonging to the second class, which are polled only once as in the case of the weighted RR.

III. Performance Evaluation

This section presents the simulation results based on the Internet access model described in the previous section. Figure 2-a shows the throughput per user for various numbers of users and *UserThinkTimes* (denoted as *UTT*) in the Internet access model where the average message length is 5 Kbytes and *InternetAccessTime* is 1 second in average. As we can see from the figure, the throughput is largely independent on the number of users when the piconet is not overloaded. Figure 2-b shows simulation results where the average message length is varied from 5 Kbytes to 1000 Kbytes while *UserThinkTimes* is fixed to 2 seconds. In contrast to the previous results, when the average message length exceeds 250 Kbytes, throughput per user decreases by up to the ratio of $1/n$ as the number of users increases to n. However, it still outperforms the fastest dial-up modem (56 Kbps) in the marketplace even when a single Internet access point is shared by seven users.

Figure 3 shows the performance in terms of delay. Figure 3-a shows the completion delay for various message lengths ranging from 5 Kbytes to 1000 Kbytes. The completion delay increases up to 50 seconds as the number of users increases when the average message length is 1000 Kbytes due to the heavy traffic. However, when the average message length is below 5 Kbytes, the completion delay is less than 5 seconds regardless of the number of users. Figure 3-b shows the measurement of user delay in the same environment. In contrast to the completion delay, variation of user delay is very small for the range of message lengths we consider. Moreover, most of the user delay is due to the *InternetAccessTime* which is fixed to 1 second in the simulation. The above results indicate that the Bluetooth-based Internet access service is feasible in terms of delay as well as throughput.

In the previous experiments, it is assumed that the



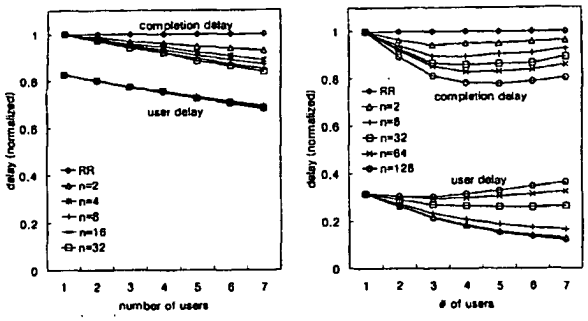
(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

Fig. 4. Completion delay in weighted RR

round-robin is the underlying scheduling policy. In the following, we present the simulation results for the other two policies relative to the round-robin assuming *InternetAccessTime* and *UserThinkTime* are 0.5 sec and 1 sec in average, respectively. Figure 4 shows the completion delay of weighted RR normalized to RR for various numbers of successive polls for each user, denoted as n. The results show that the weighted RR policy improves the completion delay by up to 15% when the traffic is low, as shown in Figure 4-a. When the traffic is heavy, the improvement is even more evident as shown in Figure 4-b, where improvement is by up to 20%. Similar results are observed for other values of parameters, although they are not included in this paper.

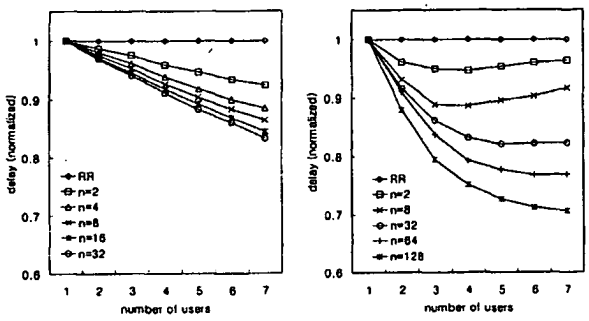
However, weighted RR increases the user delay although it improves the completion delay. Figure 5 shows the measurement of the user delay as well as the completion delay, both of which normalized as the completion delay of RR. When the load is low, increase in the user delay is marginal as shown in Figure 5-a. However, when the piconet is overloaded, the user delay increases by up to 24% compared to RR as shown in Figure 5-b. Note that the increase of the user delay is relatively small compared to the reduction in the completion delay. Overall, weighted RR policy gives performance improvement of up to 20% compared to RR.

The delay of weighted RR can be improved by giving priority to the users with download data as explained in Section II. Figure 6 shows the completion delay of the multi-level RR normalized to the delay of RR. The figure shows that weighted RR improves the completion delay by up to 25% when the load is high, and by up to 15% even when the load is low. From the results, we can notice that multi-level RR is even more efficient than weighted RR



(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

Fig. 5. User delay in weighted RR



(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

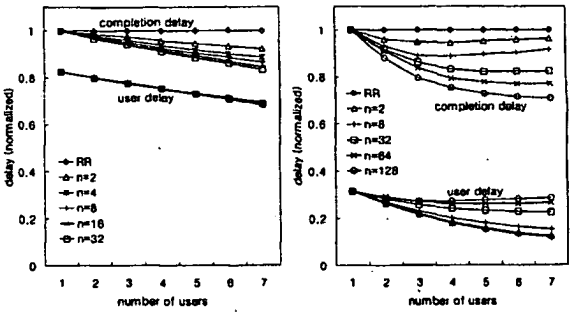
Fig. 6. Completion delay in multi-level RR

especially in an overloaded situation. The results also show that the best performance is achieved when the parameter value of n is set properly based on the transaction unit.

Figure 7 shows the results of the user delay along with the completion delay. Similarly to weighted RR, when the load is low, difference in the user delay is marginal. However, when the load is high, the user delay increases by up to 15%. Note that this number is much smaller than the case of weighted RR, where the increase is by up to 24%. The results show that the multi-level RR improves the completion delay better than weighted RR with smaller increase in the user delay. Thus, in the Internet web surfing environment with burst traffic, scheduling policies based on a transaction unit such as the weighted RR and the multi-level RR are better than policies based on a frame unit such as the pure round robin.

IV. Analysis on Inter-channel Interference

In the previous sections, a single Bluetooth unit in the Internet access point is assumed to service several users by forming a piconet consisting of all the users and the Internet access point. In this setting, all piconet members share a single Bluetooth channel under the control of the master of the piconet. A Bluetooth channel follows a unique frequency hopping sequence determined by the Bluetooth address of the master. Since different Bluetooth channels use different frequency hopping sequences, more than one



(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

Fig. 7. User delay in multi-level RR

Bluetooth channel can co-exist in a single Bluetooth cell. This gives a design choice of facilitating multiple Bluetooth units in a single Internet access point. However, multiple channels interfere with each other by occasionally using the same frequency band. Specifically, there are 79 different frequency bands defined in the Bluetooth specification [6] and the probability of two independent channels using the same frequency band at any given time is 1/79. If both channels transmit a message simultaneously when this happens, a collision occurs and eventually the message is garbled.

Of course, Bluetooth has a mechanism to detect if a message is garbled and retransmit it if so. On the receipt of a message, the recipient sends the acknowledgement to the sender, usually piggybacking on the next message. If the sender does not receive a proper acknowledgement, the message is retransmitted, usually at the next hop. Note that Bluetooth does not require a back-off mechanism to avoid repetitive collisions on retransmission, in contrast to other MAC protocols such as ALOHA since the probability of collision at the next hop is independent of the collision. This also simplifies the probabilistic model of collisions as given in the following.

We begin our analysis with the probability of a message transmitted in a channel being garbled by another channel. For simplicity, we assume that each message occupies a single Bluetooth slot and all channels are aligned each other so that the start of each slot is synchronized. Under this assumption, the probability of a message being garbled by another channel is the product of two probabilities, the probability of two channels hopping onto the same frequency band and the probability of a message being carried in a slot. The latter can be thought of as a normalized load carried over a channel including both new messages and retransmitted messages. If we denote this term as G, the probability of a message being garbled is given as $G/79$. Then the probability of a message not being garbled when there are N channels with the same characteristic, is equal to the probability of a message not garbled by any of $N-1$ channels. Hence, we get

$$P_{nocollision}(N) = (1 - G/79)^{N-1}. \quad (1)$$

In addition to the message, the corresponding acknowl-

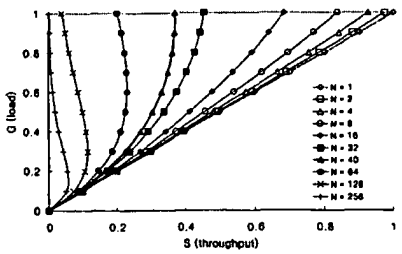
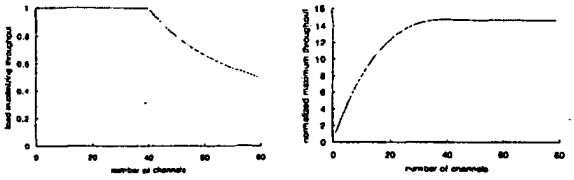


Fig. 8. Throughput characteristic



(a) Carried load (b) Maximum throughput

Fig. 9. Maximum throughput analysis

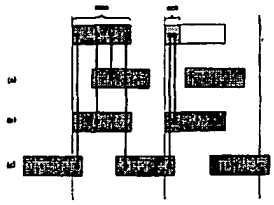


Fig. 10. Alignment of channels

edgement also needs to be transmitted without being garbled in order that a message is regarded as transmitted successfully by the sender. If we ignore the case of possible forward error correction and assume that the probability of an acknowledgement being garbled is the same as a message being garbled, the probability of a transmission being successful is:

$$P(N) = P_{nocollision}(N)^2 = (1 - G/79)^{2(N-1)} \quad (2)$$

Given the probability of a transmission being successful $P(N)$ and the carried load G, we obtain the ratio of messages transmitted successfully on a channel, i.e., *throughput*, by multiplying G by $P(N)$.

$$S = GP(N) = G(1 - G/79)^{2(N-1)} \quad (3)$$

The aggregated throughput of all of N channels, i.e., the *throughput of the Internet access point*, is given simply as S multiplied by N:

$$S(N) = NGP(N) = NG(1 - G/79)^{2(N-1)} \quad (4)$$

The relationship between S and G for various N is shown in Figure 8. The figure shows that for smaller N the throughput S increases almost linearly as the carried load G increases. However, for large N, throughput does not increase beyond certain points, similarly to ALOHA [7]. For example, when N is 128, S increases towards the maximum value until G reaches about 0.3 and drops beyond this point. Such threshold can be obtained by differentiating S by G.

$$\begin{aligned} \frac{dS}{dG} &= \frac{d}{dG}G(1 - G/79)^{2(N-1)} \\ &= (1 - G/79)^{2(N-1)-1}(1 - \tfrac{2N-1}{79}G) \end{aligned} \quad (5)$$

By setting Equation 5 equal to zero, we get $G = 79/(2N - 1)$. Since $G \leq 1$ by definition, we get $N > 40$. This implies that such threshold exists only when N is greater than 40. Thus the threshold G_{max} is represented as follows:

$$G_{max} = \begin{cases} 1 & \text{if } N \leq 40 \\ \frac{79}{2N-1} & \text{otherwise} \end{cases} \quad (6)$$

G_{max} for ranges of N is plotted in Figure 9-a. Replacing G in Equation 6 with G_{max}, we obtain the maximum throughput $S_{max}(N)$ as shown in Figure 9-b. The figure clearly shows that the maximum throughput does not improve when N increase beyond 40. From the results, we can conclude that the performance of the Internet access point improves as the number of Bluetooth units increases up to 40 although the improvement becomes smaller due to inter-channel interference.

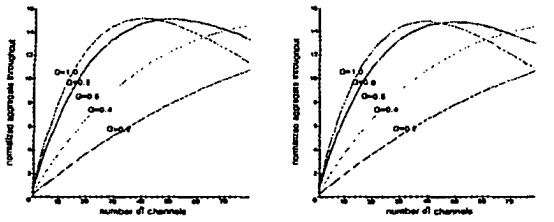
The previous analysis assumes that each message occupies a single slot and the channels are aligned with each other as shown in Figure 10-b (Case B). In this subsection, we try to relax the second assumption. When channels are not aligned as in the case shown in Figure 10-c (Case C), a message can collide by either of two messages in a channel, increasing the probability of a collision. On the other hand, Figure 10-a shows the case where the acknowledgement cannot be garbled by the other channel, improving the probability of a transmission being successful. Assuming that the three cases are generated randomly and that all the messages are carried over DH1 packets, the probability of each case is given as follows:

$$\begin{aligned} p_1 &= \frac{l-m-h}{l} = \frac{133}{625} = 0.2128, \\ p_2 &= \frac{l-m+h}{l} = \frac{385}{625} = 0.6160, \\ p_3 &= \frac{2m-l}{l} = \frac{107}{625} = 0.1712 \end{aligned} \quad (7)$$

where p_1, p_2, and p_3 represent the probability of two channels being in Case A, Case B, and Case C, respectively, and h, m, and l represent the length of the header of a DH1 packet (126 μsec), the length of a DH1 packet including the header (366 μsec), and the length of a slot (625 μsec), respectively.

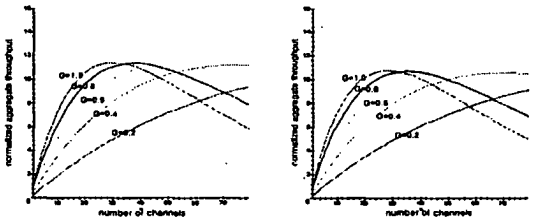
Applying binomial twice, we obtain the normalized throughput $S'(N)$ as follows:

$$\begin{aligned} S'(N) &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{N-1}{i}\binom{N-1-i}{j} \\ &\quad \times p_1^{N-1-(i+j)}NG(1 - G/79)^{N-1-(i+j)} \\ &\quad \times p_2^i NG(1 - G/79)^{2i} \\ &\quad \times p_3^j NG(1 - G/79)^{3j} \\ &= NG(1 - G/79)^{N-1} \\ &\quad \times \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{N-1}{i}\binom{N-1-i}{j} \\ &\quad \times p_1^{N-1-(i+j)}p_2^i p_3^j(1 - G/79)^{i+2j} \end{aligned} \quad (8)$$



(a) aligned-channel model (b) unaligned-channel model

Fig. 11. Comparison of both models (DH1)



(a) DH3 (b) DH5

Fig. 12. Throughput in on unaligned channels

Figure 11 compares the previous model (*aligned-channel model*) and the relaxed model (*unaligned-channel model*). The figure shows that the difference between two models is marginal. It is due to the fact that the Case A (probability: 0.2128) and Case C (probability: 0.1712) offset each other and Case B (probability: 0.6160) dominates, which is the same case as the previous model.

In the case of DH3 or DH5, the equation remains unchanged but the probability p_1, p_2, and p_3 are computed in a slightly different manner:

$$\begin{aligned}
p_1^{DH3} &= \frac{3l-m^{DH3}-h}{3l} = \frac{151}{1875} = 0.0806, \\
p_2^{DH3} &= \frac{3l-m^{DH3}+h}{3l} = \frac{403}{1875} = 0.2149, \\
p_3^{DH3} &= \frac{2m^{DH3}-3l}{3l} = \frac{1321}{1875} = 0.7045, \\
p_1^{DH5} &= \frac{5l-m^{DH5}-h}{5l} = \frac{137}{3125} = 0.0438, \\
p_2^{DH5} &= \frac{5l-m^{DH5}+h}{5l} = \frac{389}{3125} = 0.1565, \\
p_3^{DH5} &= \frac{2m^{DH5}-5l}{5l} = \frac{2599}{3125} = 0.8317
\end{aligned} \quad (9)$$

where m^{DH3} and m^{DH5} are the size of a DH3 packet (1598 μsec) and the size of a DH5 packet (2862 μsec), respectively. The results are shown in Figure 12. Since p_3 is greater compared to the case of DH1, the normalized throughput is reduced due to a larger collision probability. Note that this does not imply that the performance is reduced by using DH3 / DH5 instead of DH1, since the throughput shown in the figure is normalized by the throughput of a single channel using DH3 and DH5, respectively. In fact, the use of DH3 / DH5 improves the performance since the overhead of packetization is much less than DH1. From the results, we can notice that both models show similar characteristics and reflect the relationship between the number of channels and the resulting aggregated throughput.

V. Conclusion

Today, access to the Internet is essential in every-day life. It is hard to think of business, research, and entertainment without the Internet. Therefore, it is a requirement, rather than an option, to facilitate functionality of Internet access in hand-held devices such as PDA for *everywhere-Internet* services. Since hand-held devices require low cost and low power consumption, Bluetooth has been regarded as one of the most promising solutions for a wireless connection between the user devices and the wired network infrastructure.

In this paper, we have presented preliminary results on the performance of a Bluetooth-based access network. In the performance evaluation, we have modeled a simple Internet access scenario and simulated while varying the values of various parameters including user's think times, message lengths, and Internet access delay. The simulation results indicate that the Bluetooth-based access network provides performance of tens of Kbps to hundreds of Kbps depending on the number of users. This result is encouraging since the Bluetooth-based access network gives performance significantly better than the traditional dial-up network without the tedious wire connection. The performance can be improved if multiple Bluetooth radio units are used in the Internet access point. We have shown that the performance improves as the number of Bluetooth units increases up to 40 based on an analytical model.

As a future work, we plan to develop a more general Bluetooth simulation package including a traffic generator that reflects a variety of traffic such as ftp, e-mail, and name-card exchange. We expect the simulation package will allow us to evaluate various aspects of Bluetooth systems including issues on scheduling policies, segmentation and reassembly, parking of user devices that are largely ignored in this paper. We also expect thorough evaluation with the new simulation tool would reveal possible causes of performance bottlenecks in the Bluetooth system.

References

[1] J. Haartsen, M. Naghshineh, J. Inouye, O. J. Joeressen, and W. Allen, "Bluetooth: vision, goals, and architecture," *Mobile Computing and Communications Review*, vol. 2, pp. 38–45, Oct. 1998.

[2] P. Bhagwat, I. Korpeoglu, C. Bisdikian, M. Naghshineh, and S. K. Tripathi, "BlueSky: A cordless networking solution for palmtop computers," in *Proceedings of the 5th International Conference on Mobile Computing and Networking (ACM Mobicom '99)*, pp. 69–76, 1999.

[3] M. Albrecht, M. Frank, P. Martini, M. Schetelig, A. Vilavaara, and A. Wenzel, "IP services over Bluetooth: Leading the way to a new mobility," in *Proceedings of the 24th Conference on Local Computer Networks*, pp. 2–11, 1999.

[4] J. Haartsen, "The bluetooth radio system," *IEEE Personal Communications*, vol. 7, pp. 28–36, Feb. 2000.

[5] The Parallel Simulation Environment for Complex Systems, "http://pcl.cs.ucla.edu/projects/parsec/."

[6] The Bluetooth Special Interest Group, "http://www.bluetooth.com/techn/index.asp," Feb. 1999.

[7] M. Schwartz, *Computer-communication network design and analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[8] P. Johansson, N. Johansson, U. Körner, J. Elg, and G. Svennarp, "Short range radio based ad-hoc networking: performance and properties," in *Proceedings of the IEEE International Conference on Communications (ICC '99)*, vol. 3, pp. 1414–1420, 1999.

Performance evaluation of the Bluetooth-based public Int point

Yujin Lim Jesung Kim Sang Lyul Min Joong Soo Ma
Dept. of Mech. Eng., Seoul Univ., South Korea;

This paper appears in: **Information Networking, 2001. Proceedings. 15th Internationa**
Publication Date: 31 Jan.-2 Feb. 2001
On page(s): 643 - 648
Number of Pages: xxiii+942
Meeting Date: 01/31/2001 - 02/02/2001
Location: Beppu City, Oita
ISBN: 0-7695-0951-7
INSPEC Accession Number:7017461
Digital Object Identifier: 10.1109/ICOIN.2001.905527
Posted online: 2002-08-07 00:18:23.0

Abstract
Bluetooth has been regarded as a promising solution to an inexpensive wireless connec application of **Bluetooth** technology has been focused mainly on replacing cables betwe general wireless telecommunication such as public Internet access via a **Bluetooth**-equi expected to be one of the most popular applications in the near future. However, it is not whether the performance of **Bluetooth**-based systems is sufficient for such an applicatio preliminary results of performance evaluation of a **Bluetooth**-based Internet access poin based on simulation of an Internet access model consisting of a **Bluetooth**-based netwo number of **Bluetooth**-equipped notebook computers. The simulation results indicate that performance comparable to the fastest dial-up modem even when a number of users sha radio unit. Better performance is expected when more than one **Bluetooth** radio unit are each unit services different users concurrently. However, **Bluetooth** units in a single radi interferes with each other since the channels established by each unit occasionally collid frequency band. This paper analyzes the impact of such interference based on an analyti channel interference. The analysis shows that the performance improves as the number up to 40. We expect the proposed inter-channel interference model is useful in the desig facilitating multiple **Bluetooth** units

Index Terms
Inspec
Controlled Indexing
Internet adjacent channel interference land mobile radio notebook computers performance evaluation radio access networks

Non-controlled Indexing
Bluetooth radio unit **Bluetooth**-based public Internet access point **Bluetooth** notebook computers PDA RFI analytical model cellular phones dial-up mod frequency band hand-held devices inexpensive wireless connection intercha interference model performance evaluation simulation results

Author Keywords
Not Available

References

No references available on IEEE Xplore.

Citing Documents

1 An improved packet collision analysis for multi-Bluetooth piconets considering freque time effect, Ting-Yu Lin; Yen-Ku Liu; Yu-Chee Tseng
Selected Areas in Communications, IEEE Journal on
On page(s): 2087- 2094, Volume: 22, Issue: 10, Dec. 2004
Abstract | Full Text: PDF (672)

2 Performance evaluation in Bluetooth dense piconet areas, Mazzenga, F.; Cassioli, D. Loreti, P.; Vatalaro, F.
Wireless Communications, IEEE Transactions on
On page(s): 2362- 2373, Volume: 3, Issue: 6, Nov. 2004
Abstract | Full Text: PDF (784)

View Search Results | Next Article

 By Angus Robinson, Anritsu

THE IMMINENT MARKET EXPLOSION OF BLUETOOTH WIRELESS INTERFACES ON PRODUCTS SIGNALS AN RF MEASUREMENT FIRST FOR MANY TEST ENGINEERS.

On your marks for testing Bluetooth

THE BLUETOOTH WIRELESS STANDARD is coming into its own, and hundreds of millions of Bluetooth-enabled products will ship by the end of 2002. The Bluetooth technology will be self-contained within many products; for others, it will be an addition in the form of a PC Card that plugs into a mobile device or a dongle that plugs into a desktop system's RS-232 or parallel-printer port. As Bluetooth becomes ubiquitous, you'll find yourself having to test Bluetooth devices at the protocol-stack and RF levels.

A complete Bluetooth module comprises a radio transceiver, a baseband-link controller, and a link manager (**Figure 1**). The baseband-link controller connects the radio hardware to the baseband-processing and physical protocol layer. The link manager performs high-level protocol activities such as link setup, authentication, and configuration. Application-layer software sits above the link manager.

A Bluetooth implementation can operate as a single-chip (integrating radio and protocol stack) or multi-chip (separate radio and protocol ICs) design. Cellular-phone manufacturers may wish to combine the Bluetooth baseband function in the same device as the GSM (Global Standard for Mobile Communication) baseband function and, therefore, will want a separate Bluetooth radio. A digital camera manufacturer is more likely to select a single-chip design to simplify assembly.

RF, PROTOCOL, AND PROFILES

Three aspects of a Bluetooth module need testing: RF, protocol, and profiles. You can perform many of the RF measurements with standard test instruments such as spectrum analyzers with vector demodulation, transmitter analyzers, power meters, digital-signal generators, and BERTs (bit-error-rate testers).

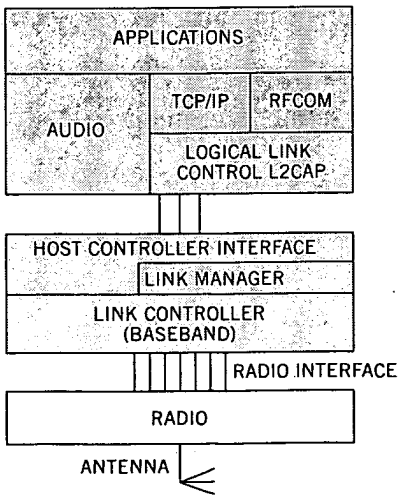
Some of the measurements, however, require the radio to form a standard Bluetooth radio-link connection with the test instrument and the test instrument to have some control over the equipment under test (**Figure 2**). For these tests, the test system must be able to support the Bluetooth protocol to make the link. Consequently, you can expect instrument vendors to develop new test instruments similar to the integrated radio test sets for digital cellular radio.

For protocol tests, you can use protocol sniffers that monitor and display data moving between Bluetooth devices. You can also use products such as the EBDK (Ericsson Bluetooth Development Kit) (**Reference 1**). Ericsson will soon release a version of the EBDK, known as a Blue Unit, that will include software for use during early qualification testing. Eventually, some companies should develop complete reference test systems.

A *profile* is the application level protocol that makes a device perform its functions as the user expects. All Bluetooth devices that claim to offer a given functionality must use the appropriate profile from the Bluetooth specification. This requirement ensures interoperability between common devices from multiple vendors.

For example, the LAN access profile defines a data connection between a DT (data terminal) and a LAP

Figure 1



A complete Bluetooth module comprises a radio transceiver, a baseband-link controller, and a link manager.

This article appeared in Test & Measurement World/ September 2000.

(LAN access point). The profile defines the following services and connection states for the application layer: initialization of LAN access service, shutdown of LAN access service, establishment of LAN connection, loss of LAN connection, and disconnection of LAN connection.

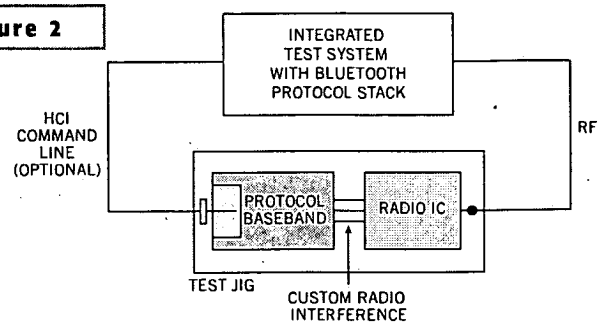
Until a reference test system is available, early Bluetooth profile testing will require product-to-product interoperability testing. To facilitate this, a series of "Unplugfests" have been arranged by the Bluetooth Special Interest Group (www.bluetooth.com). At these meetings, companies with functional products can test product interoperability against products from other suppliers.

RF AND SINGLE-CHIP MEASUREMENTS

The Bluetooth radio specification outlines the performance requirements for the radio and the test to confirm conformance (**Table 1**). To measure the performance of a Bluetooth module, the test instrumentation must be able to establish a Bluetooth link with the EUT (equipment under test). It can then put the EUT into test mode. Test mode is a mandatory feature of a Bluetooth module in which the EUT can enter a loopback mode or can disable frequency hopping for making BER measurements, for example. A Bluetooth test system should also be able to disable hopping, set specific frequencies for tests, and control the transmit power level.

During design and development, you'll want to test the radio in isolation from the protocol stack that controls it. In these cases, you need to control the radio so you can set frequencies and levels to make raw transceiver measurements. You can then feed the radio output directly into a spectrum analyzer with vector signal analysis or into a transmitter analyzer and power meter to make the measurements (**Figure 3**). The measurements you can make will vary according to the control that the radio IC manufacturer provides over the IC connectors. The radio will have inputs controlling data in and out, Tx on, clocks, and supply voltage. Without a protocol stack, the radio may not function at all.

Figure 2



Some Bluetooth tests require the test instrument to support the Bluetooth protocol to form a standard Bluetooth radio-link connection.

Many radio designs permit you to feed PRBS (pseudo-random bit sequences) into the transmitter modulator and use manufacturer-specific control lines to force the radio to transmit continuously at one frequency. Doing so enables you to make frequency, power, and modulation measurements as well as output spectrum measurements. As an alternative method to radio IC testing, you can build a test jig that holds the radio IC and has a protocol stack built onto it. This method allows for comprehensive testing and simulates the integrated module approach outlined below.

There is no standard for the interface between a protocol stack and a radio, typically known as the radio interface. You must use a test fixture to hold the radio IC and provide it with some baseband control to give Bluetooth functionality. This arrangement lets you test radio ICs with a Bluetooth link and baseband control.

The connection between the radio and the test instrument vary depending upon the Bluetooth device implementation. Some radios or integrated modules have printed antennas as part of the design. In this case, you may only be able to make a connection over the air to an antenna on the test instrument input. If you use this approach, then you must characterize the path loss for each of the 79 Bluetooth frequencies. If the radio IC has an RF output connector, a direct connection to the test instrument simplifies calibrated power and sensitivity measurements. Even if you use a direct connection, though, you should measure and correct for the path loss at each frequency.

TESTING OEM BLUETOOTH PRODUCTS

OEMs buying commercial Bluetooth chip sets still need to test. Inevitably, packaging can influence the performance of a finished product because of the antenna's position as well as other internal electronics. In a mobile phone, the other

HOW TO QUALIFY PRODUCTS WITH BLUETOOTH INTERFACES

No product may be sold as "Bluetooth enabled" without first demonstrating compliance with the Bluetooth specification. You can refer to the rules laid out in the Bluetooth Qualification Program when you check compliance. Qualification is essential for ensuring that consumers have a good experience with Bluetooth-branded products. You must make sure that interoperability between products supplied by different manufacturers is guaranteed.

To obtain qualification, a manufacturer must first become a Bluetooth member by signing the adopters agreement. Two additional bodies help qualify a product: a BQTF (Bluetooth Qualification Test Facility) and a BQB (Bluetooth Qualification Body).

A BQTF is an accredited organization with the skills and equipment to test a product based on the Bluetooth specification. A BQTF may choose not to offer qualification for every aspect of the Bluetooth standard. For example, many profiles are limited to a few specific applications, and some aspects of the Bluetooth specification are optional. The BQTF performs measurements on behalf of the manufacturer on the appropriate radio, protocol, and profiles for the equipment under test. The BQTF prepares a test report that forms part of a compliance folder that is submitted to the BQB.

The role of the BQB is to review all submitted documentation and ensure that all the appropriate tests have been performed and passed satisfactorily. If all is well, the product is listed as Bluetooth qualified and may be sold as Bluetooth enabled.

electronics will, by definition, include an interfering radio transceiver. Similarly, PCs have high-speed clocks or noisy buses that can degrade module sensitivity.

In the OEM production environment, the tests have to validate performance in the shortest possible time. Production engineers need to select the subset of the conformance test specifications that are appropriate for their products' requirements.

To confirm that the device will operate

TABLE 1—THE BLUETOOTH RADIO SPECIFICATION INCLUDES CERTAIN RADIO TESTS TO CONFIRM CONFORMANCE

Transmitter tests

Test	Limits	Hopping	Test mode	Loopback (or Tx)	Payload	Packets	Certification script
Output power	20 dBm, 4 dBm, 0 dBm	On	On	Loopback	PRBS 9	Longest supported	Low/middle/high frequencies over one packet
Power control	Power class dependent +20 to −30 dBm	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over one packet
Modulation characteristics	140 kHz ≤Δf ≤175 kHz Δf≥ 115 kHz Δf2/Df1 ≥ 0.8	Off	On	Loopback	10101010 & 11110000	Longest supported	Low/middle/high frequencies over 10 packets
Initial frequency	±75 kHz	Off and on	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over 10 packets
Frequency drift	DH 1 ±25 kHz DH 3 ±40 kHz DH 5 ±40 kHz Overall: ≤ 4000 Hz/ 10 µs	Off and on	Tx	Loopback or Tx test	1010...	Longest supported	Hopping off low/middle/high over 10 packets, all supported lengths. Hopping on all frequencies over 10 packets, all supported lengths
Power density	<20 dBm per 100 kHz EIRP	On	On	Loopback	PRBS 9	Longest supported	One minimum peak power
TX output spectrum frequency range	Country-dependent (e.g., Euro/US 2.4 to 2.4835 GHz)	Off	On	Loopback	PRBS 9	DH 1	−80 dBm/Hz EIRP
TX output spectrum 20 dBm	$\Delta f = \lvert f_h - f_l \rvert$ ≤1.1 MHz	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies
TX output spectrum adjacent channel	≤−20 dBm ACP other conditions apply	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies with conditions
Out of band spurious	30 MHz to 12.5 GHz	Off	On	Loopback	PRBS 9	DH 1	Maximum power, high low frequency

Receiver tests

Test	Limits	Hopping	Test mode	Loopback (or Tx)	Payload	Packets	Certification script
Sensitivity single-slot	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over 1,600,000 returned payload bits. Dirty transmitter
Sensitivity multi-slot	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 5 (or DH 3 if DH 5 not supported)	Low/middle/high frequencies over 1,600,000 returned payload bits. Dirty transmitter
Maximum input	−20 dBm at receiver input	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over 1,600,000 returned payload bits
C/I performance	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Two generators required: BT and BT modulated
Blocking performance	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Two generators required: BT and CW
Intermodulation	BE≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Three generators required: BT, BT modulated, and CW

Notes: ACP (adjacent channel power); BT (Bluetooth); DH (Data High); EIRP (equivalent isotropically radiated power); PRBS (pseudo-random bit sequence); TX (transmitter).

over the Bluetooth specification's 10-m range, engineers will still need to measure sensitivity and power levels. The conformance specification requires engineers to measure receiver sensitivity as a BER of more than 1,600,000 bits at three frequencies. This test alone would take at least 25 sec using standard single-slot DH1 packets and so, in practice, the test will measure fewer bits even at a reduced number of frequencies.

In addition to RF measurements, OEMs should perform a functional test. In the case of a Bluetooth-enabled digital camera, for example, functional testing can include sending an instruction over the Bluetooth interface to activate a shutter release with flash. Engineers would need to create this command at a high level in the protocol stack (typically the application level), and the test equipment would need to package the command into the Bluetooth format. Validating the camera's response to a high-level command would give the manufacturer confidence that the interface was functioning correctly, although it is not necessarily a guarantee of robust performance.□

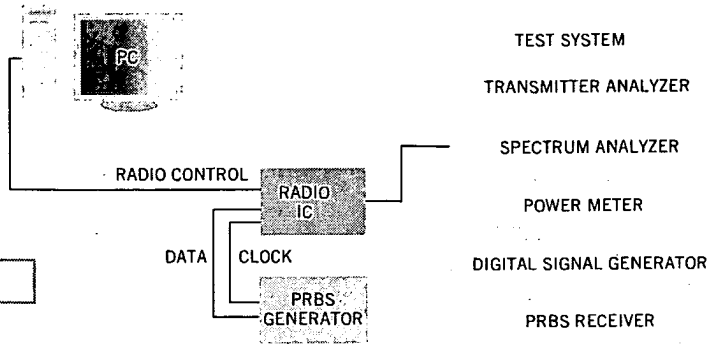


Figure 3

In development, you make raw transceiver measurements of a Bluetooth radio IC in isolation from the protocol stack.

REFERENCE

1. Available from Symbionics. www.symbionics.co.uk/solutions/bluetooth/Bluetoothkit.shtml.

AUTHOR BIOGRAPHY

Angus Robinson is product marketing manager for RF and microwave test instruments with Anritsu, Stevenage, UK. He joined Anritsu in 1998 having previously worked with Marconi Instruments (now IFR) after receiving a Bachelor of Engineering degree in electronics at Liverpool University in 1982.

BACKGROUND TO BLUETOOTH

As a standard for wireless communication between multiple devices, Bluetooth supports voice and data. In 1994, Ericsson began to develop the standard at its Lund site in Sweden (**Table A**). The project was initially called MC Link (multicommunicator link). In 1997, Ericsson approached other companies with a mutual interest in defining an open standard for a wireless link. In February 1998, five promoters—Ericsson, IBM, Intel, Nokia, and Toshiba—formed the SIG (Special Interest Group) to promote the standard, which they renamed Bluetooth.

The Bluetooth SIG announced the standard on May 20, 1998. Two years later, more than 1800 companies had joined the consortium as adopters of the technology. The consortium's objective is to create a de facto short-range wireless communication standard that all companies could use. In the autumn of 1999, the promoter group was expanded to nine companies, adding 3Com, Lucent Technologies, Microsoft, and Motorola. Although today the Bluetooth SIG standard is "owned" by the promoter group, it is expected that the standard will become an IEEE standard (802.15) this year and remain fundamentally the same.

Bluetooth offers wireless communication between one or more devices over a 10-m range with a maximum gross data rate of 721 kbps in the unlicensed 2.4-GHz ISM band. The purpose of the technology is to offer a low-cost, simple-to-use alternative to wired connections. As such, the potential user base is large and varied.

The first adopters of Bluetooth are expected to be mobile- phone manufacturers with wireless headsets. Mobile phones could also interface with a Bluetooth-enabled PCs to exchange files or e-mail. A PC could have a wireless mouse and keyboard, and office printers could become Bluetooth enabled so that wireless printing is possible in any location. Other early adopters of the technology include PDAs (personal digital assistants), LAN access points, digital cameras, and security-access cards.

TABLE A—GENERAL BLUETOOTH RADIO SPECIFICATION

Parameter	Specification	Comments
Link type	SCO (Synchronous connection oriented)	Point-to-point, full duplex link, circuit switched, symmetric with fixed-interval slot reservation
	ACL (Asynchronous connectionless link)	Momentary connection, packet switching asynchronous with polling access
Frequency	ISM band, 2.402 to 2.480 GHz, 79 channels—1-MHz spacing, hopping at 1600 hops/s	Now a common frequency plan including France, Spain, and Japan
Modulation	2 FSK with 0.5 bandwidth time Gaussian filter, peak deviation 175 kHz, modulation index 0.28 to 0.35	
Data rate	1 Mbaud gross, maximum net data rate 721 kbps, voice channel 64 kbps	Actual data rate depends upon packet length, use of error correction, and encryption
Transmitter power	Class 1, P >0 dBm, Class 2, P −6 to +4 dBm, Class 3, P <0 dBm	

Personal Area Network: A Bluetooth Primer

by Albert Proust
11/03/2000

Related:

Anyone who's ever wrestled with an octopus of black and beige cords coming out of the back of their laptop computer can appreciate

the value of Bluetooth. Bluetooth is an emerging standard and a spec for (in the words of the Bluetooth SIG [special interest group]): "small-form factor, low-cost, short range radio links between mobile PCs, mobile phones and other portable devices." In other words, it's a wireless connection between PCs, peripherals, and portables that will let the devices share and synch information, without having to make a physical connection.

An Introduction to Lucent's Wavelan Wireless LAN Card
Connecting PCs to Apple's Wireless Airport
Affordable Wireless LAN Using Airport
Wireless News - from Meerkat
Previous Features ▶
More from the Wireless DevCenter ▶

Bluetooth's original backer is the Swedish mobile phone maker, Ericsson, which named the technology for a 10th-century king of Denmark, Harald Blåtland, who unified the Danes and Norwegians. The spec's origins date back to 1994, when four companies joined Ericsson to develop the technology: Nokia, Toshiba, Intel, and IBM. Today, the Bluetooth SIG includes nearly 2,000 companies, and prototype devices are beginning to make their way into the marketplace.

Although the idea behind Bluetooth (wireless communication between devices) has been around, it's the momentum behind this standard and the agreement among hundreds of vendors and manufacturers that has brought it to the verge of becoming a reality. The power of the Bluetooth vision begins to really emerge when you consider a world of devices intelligently connected and carrying much of their communication load automatically.

Imagine having a Bluetooth-enabled phone or PDA on you. As you approach your home, you're able to control lighting, heating, even locks with your PDA. As you enter your home, you can use the same device to turn on the television set or the stereo system. Meanwhile, your refrigerator takes the initiative to update your shopping list. As you can see, the full picture includes a whole new level of automation where devices and appliances are programmed to communicate important information to each other, with or without human intervention.

How it works

Bluetooth uses the radio waves located in the frequency band of 2.4 GHz (2400 to 2483.5 MHz), an increasingly popular (and crowded) slice of the spectrum. In this band, Bluetooth transmits voice and data at flows lower than 1 megabit per second.

Bluetooth devices can function in two modes:

- circuit switched (the most common mode for voice communications, on land and wireless digital networks), and
- packet switched (the mode for Internet data, as well as for higher bandwidth mobile communication systems on the horizon, like GPRS [General Packet Radio Service]).

A device can use either one or both of these modes. In packet switched mode, connection is asynchronous with a rising flow of 57.6 Kbps to 721 Kbps. In the second case, connection is synchronous with a flow of 64 Kbps.

Piconet and Scatternet

A Bluetooth network (known as Piconet) can allow the interconnection of eight devices in a radius of 10 meters. This network can be fixed or provisional (a mobile or transitory network). In a Piconet, the Master seeks the devices in its entourage by emitting requests (broadcast). The slave answers with its identification number.

As many as 10 Piconets can overlap to form a Scatternet, linking up to 80 Bluetooth appliances. Beyond this, the network saturates. Indeed, only 79 transmission channels are employed by the Bluetooth protocol, a limit based on the frequency.

By default, Piconets transmit up to 10 meters (about 30 feet). However, you can increase it to 100 meters by increasing the power output of 100 mW (milliwatts), as opposed to the 1 mW of default Bluetooth. However, compared to GSM (Global System for Mobile communications), which consumes between 1.5 and 2 Watts, this is still a weak signal. Manufacturers are working to make Bluetooth devices that adapt to the necessary proximity, so as not to consume more energy than is necessary.

The Personal Area Network

Bluetooth isn't designed to compete with wireless local area networks. Even its close-range throughput of 1 Mbps doesn't compare with the 11 Mbps that the emerging standard for wireless LAN, IEEE 802.11, offers.

Instead, Bluetooth's promoters are positioning it as the technology for the Personal Area Network (PAN), and are targeting appliances that don't require large flows -- like printers, personal computers, and mobile phones. One concept that's been put forward is the mobile PAN: a communication device clipped to your belt could contain a GSM transceiver that communicates with the wider world. Meanwhile, the same device has a Bluetooth transceiver that communicates with your headset (replacing your mobile phone), your PDA, your MP3 player, allowing all these devices to communicate with each other and the larger world.

Bluetooth Sources

Bluetooth SIG
This site has details about the Bluetooth special interest group, its meetings, and its members. There's also a cartoony Flash animation on the history of Harald Blåtland and the Vikings.

Ericsson
Ericsson's T36 is the first Bluetooth-enabled phone.

Hung Up On Gadgetry
ZDNet's eWeek

Since it is not a very expensive technology (between $5 and $20 per

chip), it can easily be placed in many devices. Also, Bluetooth doesn't require an access point, unlike the traditional radio operator networks. It's well suited for mobile devices, since it can join a local Piconet quickly, as soon as the two devices are in a sufficient perimeter.

And unlike infrared networks (like two Palm computers beaming each other), Bluetooth doesn't require you to align objects for them to communicate.

Towards the intelligent whole

Although most Bluetooth devices are still at the prototype stage, Ericsson has delivered the first Bluetooth-enabled phone, the Ericsson T36. This GSM phone uses Bluetooth to communicate with the handset. Thus, a user could wear the headset and chat away while the handset was stashed away in a briefcase. Of course, since the power of the technology is in the network, a single Bluetooth device without others to communicate with is ... well, a start at least.

Still, the momentum behind this de facto standard suggests a ripe market ahead: Cahners In-Stat Group predicts 1.4 billion Bluetooth-compatible devices by 2005. That's a conquest King Harald could be proud of.

Albert Proust is co-authoring a book on WAP for O'Reilly Media, Inc..

Related:

An Introduction to Lucent's Wavelan Wireless LAN Card

Connecting PCs to Apple's Wireless Airport

Affordable Wireless LAN Using Airport

Wireless News - from Meerkat

Discuss this article in the O'Reilly Network Wireless Forum.

Return to the Wireless DevCenter.

ONLINE feature

Bluetooth Bites Information Retrieval

Maryellen Mott Allen

ONLINE, May 2001

Internet world is replete with buzzwords, trends, and rumors about the latest technologies coming to the fore. Ask anyone these days what the new darling of the Internet is, and they will tell you without a doubt that it's wireless technologies. Online journals, newsgroups, discussion forums, and print publications concerning the Internet are engaged in a furious exchange of opinions in which the pros and cons of various wireless data transmission protocols are alternately praised and reviled. One such protocol, simultaneously adored and despised, is Bluetooth (http://www.bluetooth.com).

According to Cahners In-Stat Group (http://www.instat.com) in its July 2000 report, "Bluetooth 2000: To Enable the Star Trek Generation," the manufacture of Bluetooth-enabled equipment will exceed one billion units by 2005, and the market will be worth some $5 billion. Frost & Sullivan (http://www.frost.com) is equally optimistic. It forecasts global shipments of Bluetooth-enabled products to reach over 11 million units in 2001 and predicts $2.5 billion in revenues.

The name itself is enough to start you wondering. When discussing the dry topic of data transmission protocols, we are used to throwing around terms such as TCP/IP, RIP, or PPP. Whether or not we can decipher the acronym, the terms have been used so often that most people have some idea of what we are talking about. But mention Bluetooth to the uninitiated and you're likely to be met with puzzled looks.

A BIT ABOUT BLUETOOTH

Bluetooth, named after the 10th century Viking king Harald Bluetooth, is a de facto data transmission standard developed by Ericsson and backed by a host of other technology companies that make up the Bluetooth Special Interest Group (including Ericsson, Intel, Puma Technology, Microsoft, Motorola, Nokia, 3Com, Lucent Technologies, and Toshiba). Bluetooth employs the unlicensed radio frequency (RF) portion of the electromagnetic spectrum in the 2.4-2.4835 gHz range. More simply, Bluetooth is a low-power, spread-spectrum technology that uses frequency-hopping to ensure speedy, short-range wireless data transfer of up to 720 kbps at a range of 30-100 feet. Intended as a replacement for the short-range data cable, Bluetooth's original conception was as a wireless peripheral interface, linking print- ers, monitors, keyboards, and other peripheral devices to the CPU. However, the potential for the technology is much greater, offering the ability to link devices of different types, forming instant workgroups across multiple platforms in a seamless fashion. So how does Bluetooth work?

The Bluetooth standard is a complex conglomeration of protocols arranged in a protocol stack, which, when diagramed, somewhat resembles a Dagwood-style sandwich. To put it more simply,

Bluetooth relies upon radio frequencies to transmit data, providing a universal bridge between devices on a network, or between devices from the outside and an existing network using a combination of circuit and packet-switching technologies. Bridges have the ability to link together different types of networks because a bridge delivers data based upon the MAC (Medium Access Control) address that is hard-coded into the network hardware by the factory that manufactures it, and it is unique for every device. This differs from other network devices, such as routers, that deliver network data based upon routable protocols such as TCP/IP. Because bridges operate at a lower level of the protocol stack, standards like Bluetooth can link unlike devices within a local network. The data is, like all network traffic, divided into packets. However, because Bluetooth is designed to work in the RF environment–a very noisy portion of the electromagnetic spectrum–it employs the use of shorter packets for transmission, and combines this with fast frequency-hopping to ensure a fairly robust connection.

When a message is sent over a Bluetooth connection, each packet is transmitted on a different frequency within a range of 2.4-2.4835 gHz. After the first packet is transmitted, the Bluetooth controller hops to a different frequency before sending out the next packet. The determination of which frequency within the given range will be used to start packet transmission is semi-random and is controlled by the Bluetooth Radio portion of the protocol stack. This frequency remains fixed for the duration of each packet. However, once the initial transmission frequency is chosen for the first packet, the remaining frequencies are cycled through for each subsequent packet in a determined fashion. This is known as the phase.

BLUETOOTH IS SOCIAL

When two or more Bluetooth-enabled devices are within range of each other, they automatically start communicating. Each Bluetooth-enabled device periodically broadcasts an inquiry message to see if there are any other Bluetooth devices in the area. If there is a response from another device, the originator of the inquiry message becomes the Master unit and the responder becomes the Slave. This arrangement is completely dependent upon which device is the first to send the inquiry message out when it comes within range, and does not resemble a client/server relationship. After this initial contact, the Master sends the slave(s) information about how they will communicate (i.e. the initial frequency and phase). These ad hoc networks are called Piconets. Collections of Piconets form Scatternets. Bluetooth supports point-to-point as well as point-to-multipoint links depending upon the number of devices within communication range. In each case, the connections are peer-to-peer.

Bluetooth can send data along four different channels simultaneously. Included is an asynchronous data channel (meaning packets are not sent out in pre-selected timeslots governed by the processor clock, but rather whenever there's an opportunity), and up to three simultaneous synchronous voice channels, or a combination channel, simultaneously supporting asynchronous data and synchronous voice. Each synchronous voice channel can support transfer rates of 64 kbps, and the asynchronous channel can support a bi-directional (two-way) asymmetric link of up to 721 kbps in either direction while permitting 57.6 kbps in the return direction.

So what does all of this have to do with libraries and information professionals? A lot, if the push to promote the standard is successful.

WILL LIBRARIES DEVELOP A TASTE FOR BLUETOOTH?

Imagine an environment where users enter the library with their personal digital assistants (PDAs)

and are instantly connected to the network. They can pull up their records, search proprietary databases and the Web, check email, download information, and even check out materials from the stacks without ever having to stop at the checkout desk. All of these things are theoretically possible using the Bluetooth standard. Bluetooth-enabled networks have the potential to allow libraries to offer the kind of value-added services often held up as the holy grail of mission statements.

In Ken Varnum's insightful article entitled "Information at Your Fingertips: Porting Library Services to the PDA," appearing in the September/ October 2000 issue of ONLINE, he illustrates how Ford Motor Company's corporate library is taking the initiative in porting library services to handheld devices, such as the PDA. The article demonstrates precisely how the engaged and aware information professional is finding new avenues for reaching out to users. Varnum explains how the library has effectively employed a third-party software package to translate Web pages into PDA-friendly files that users can download and take along with them on trips away from the office. This technology allows users to view documents, check and respond to email messages, and even fill out online forms. The drawback, of course, is that the user must return to the office and interface the PDA with a PC in order to synchronize the devices and actually transmit the email, form, etc. The PDA itself is not connected online and does not allow for real-time operation.

Now, take that scenario and replace the old PDA with a new Bluetooth-enabled PDA. Then suppose traveling users find themselves at a hotel, airport, or library with a Bluetooth-enabled network connected to the Internet. Our users can now not only respond to email messages, but send them as well. They can conduct research and download updated documents. They can even use their PDAs to make phone calls. In short, the user has been given the type of information access that would have been unimaginable just a few years ago.

BLUETOOTH IN TRADITIONAL SETTINGS

Even the more traditional academic or public libraries could reap the benefits of a Bluetooth-enabled network. Librarians at the reference desk assisting patrons in their research could directly beam the results to the patron's PDA, avoiding the cost of printing out the documents. Patrons with Bluetooth enabled devices connected to the library network could work together in ad hoc user groups, sharing information electronically. Library instructional sessions could be greatly facilitated by turning a patron's personal device into an instant workstation, eliminating the need for the library to purchase and maintain expensive computer labs. Network printers could be placed throughout the library for those who wish to obtain printouts without the restrictions that come with traditional network cables or the line-of-sight problems that accompany devices using infrared. A prototype description of Swedish rail traffic in the future (http://www.swedetrack.se/usblue2.htm) proposes placing Bluetooth units at public libraries close to train stations to be used as passenger interfaces.

LEXIS-NEXIS has teamed up with a primary developer of the Bluetooth Standard and member of the Bluetooth SIG, Red-M, a subsidiary of the Dutch company Madge Networks, to launch m-news. With content from LEXIS-NEXIS, m-news will provide subscribers with email updates and hypertext links to stories located on the Red-M Web site that detail Bluetooth industry developments. Registration is required but the service is free (http://www.redm.com/aboutred/newsroom/content.asp?Article=21).

And if all of that sounds too good to be true, you may be right.

BLUETOOTH MAY HAVE PROBLEMS AT ITS ROOT

The potential for Bluetooth is enormous, but its road to acceptance has been rife with criticisms and obstacles, some of which are substantial indeed. First there is the issue of cost. The original model developed by the Bluetooth SIG in 1998 called for a figure of $5 to integrate Bluetooth radio transceivers and link-level controllers into hand-held devices. The reality has been more in the range of $50 for the components, or slightly less if purchased in large quantities. And this is the price paid by the original equipment manufacturers (OEMs). Undoubtedly, by the time the device reaches the market, the price will be considerably higher. This has effectively put it out of the reach of most consumers. The inflated prices, however, are not the result of corporate greed, but more the unpredictable consequence of the technical difficulty of producing the chips and the resulting size of the chip being somewhat larger than anticipated.

Another significant problem is that of interference. As Joe Wilcox, staff writer for CNET News.com notes in his September 15 article, "As Bluetooth Nibbles, Competition Lurks" (http://news.cnet.com/news/0-1003-200-2784702.html), the issue of frequency interference could potentially harm Bluetooth a great deal. Because Bluetooth uses the unlicensed radio portion of the electromagnetic spectrum, transmissions in that band must compete with industrial microwave ovens, stadium lights, garage door openers, cordless telephones, and virtually any wireless household appliance, even baby monitors. More detrimental to the long-term health of the Bluetooth standard, however, is its interference with two other wireless standards using the same 2.4 gHz band that are older and more established: that of the 802.11B and the HomeRF. While these standards lack the catchy name and media hype that Bluetooth currently enjoys, 802.11B allows portable electronic devices to connect to an existing network at distances up to 300 feet rather than the paltry 100 feet that limits Bluetooth. Additionally, HomeRF recently won a ruling from the FCC that allows it to expand its bandwidth to 5mHz (up from 1mHz), effectively increasing transmission speeds to 10 Mbps. While there is room for these standards to operate in a complementary fashion within a wireless network environment, another challenge rears its head: compatibility.

The bickering over who has control over various portions of the spectrum gets even uglier when dealing with the unlicensed portion that Bluetooth and other standards exploit. There is little that the FCC can do to regulate this area and still preserve the integrity of maintaining an open portion of bandwidth. To get around some of the interference problems, both HomeRF and Bluetooth use frequency-hopping. In contrast, the 802.11B standard has developed around a direct-sequence model in which only one frequency is used for transmission. The use of these two different methods introduces incompatibility between devices. A frequency-hopping Bluetooth device and a direct-sequence 802.11B device would not be able to communicate with each other.

Add to that the security problems inherent in the Bluetooth standard, and the outlook appears grim. There is a security layer within the specification for Bluetooth, but by all accounts, it is easily misunderstood and prone to confusion. As Wilcox reports, an analyst with Gartner, a leading Internet industry consulting firm states, "Bluetooth is a disaster waiting to happenÉthe specs cover (security), but unless you know what you're doing, it's possible to implement the spec in such a fashion (that) you aren't doing anything worthwhile."

IT TAKES TWO TO TANGO

The rollout of Bluetooth-enabled devices has experienced many delays as a result of some of these issues. When the Bluetooth SIG was first formed, the company fully expected to see widespread implementation by Christmas of 2000. Unfortunately, it has been only recently that Bluetooth-equipped devices have begun to trickle out of the development labs and into the hands of

consumers. Even so, there are not enough Bluetooth products around to make it a worthwhile investment. One Bluetooth device is no good–you really need at least two to play. Yet it is important to bear in mind that these problems are not insurmountable. Dozens of companies have significant resources sunk into the development of Bluetooth. To abandon it would represent a tremendous loss of time and money. While the future of the technology is by no means secure (there are a lot of kinks to work out), you can rest assured that much effort will go into its continued development and promotion. We'll see within the next few years if Bluetooth will sink its teeth into the information industry, or bite itself in the butt.

Harald Bluetooth

Harald Bluetooth, son of Denmark's first king, Gorm the Old, from whom the present Danish queen, Margarethe II, traces direct descent, did not have blue teeth. Instead, the name refers to a great man with a dark complexion. The Danish word for blue, *blå,* also meant dark and the words for man, mand, and tooth, *tand,* sound much the same. Harald Bluetooth is credited with Christianizing Denmark, Norway, and parts of Sweden and with uniting the countries into one kingdom. At the time of his rule, somewhere between 940 and 980 AD, southern Sweden was part of Denmark. In southern Sweden is the city of Lund, which is where Ericsson developed the Bluetooth technology. According to Ericsson in its discussion of Harald Bluetooth (http://bluetooth.ericsson.se/bluetoothf/beginnersg/default.asp?page=2), "One of his skills was to make people talk to each other....," hence the choice of Bluetooth to name this communications standard. If you're ever in Jelling, Denmark, you can view the rune stone Harald Bluetooth raised in honor of his parents. There's another, vastly newer, rune stone dedicated to Bluetooth himself. It's outside Ericsson's Mobile Communications office in Lund.

Maryellen Mott Allen *(mallen@lib.usf.edu) is Instructor Librarian, University of South Florida Tampa Campus Library.*

Comments? Email letters to the Editor at marydee@infotoday.com.

[infotoday.com] [*ONLINE*] [Current Issue] [Subscriptions] [Top]

Performance Evaluation of the Bluetooth-based Public Internet Access Point

Yujin Lim†, Jesung Kim, Sang Lyul Min, and Joong Soo Ma‡

Department of Mechanical Engineering, University of Seoul, Seoul, Korea†
School of Computer Science and Engineering, Seoul National University, Seoul, Korea
Information and Communications University, Taejon, Korea‡

yujin@uos.ac.kr jskim@archi.snu.ac.kr symin@dandelion.snu.ac.kr jsma@icu.ac.kr

Abstract— **Recently, Bluetooth has been regarded as a promising solution to an inexpensive wireless connection. Although initial application of Bluetooth technology has been focused mainly on replacing cables between hand-held devices, general wireless telecommunication such as public Internet access via a Bluetooth-equipped device is expected to be one of the most popular applications in the near future. However, it is not well understood whether performance of Bluetooth-based systems is sufficient for such an application. In this paper, we present preliminary results of performance evaluation of a Bluetooth-based Internet access point. The evaluation is based on simulation of an Internet access model consisting of a Bluetooth-based network access point and a number of Bluetooth-equipped notebook computers. The simulation results indicate that Bluetooth provides performance comparable to the fastest dial-up modem even when a number of users share a single Bluetooth radio unit. Better performance is expected when more than one Bluetooth radio unit are employed such that each unit services different users concurrently. However, Bluetooth units in a single radio range (about 10m) interferes each other since the channels established by each unit occasionally collide at the same frequency band. This paper analyzes the impact of such interference based on an analytical model of inter-channel interference. The analysis shows that the performance improves as the number of channels increases up to 40. We expect the proposed inter-channel interference model is useful in the design of systems facilitating multiple Bluetooth units.**

I. Introduction

Interconnection of consumer devices has been increasingly important as the use of hand-held devices such as PDA's and celluar phones becomes increasingly popular. In the past, these devices were usually connected via a serial cable that requires physical wiring between two devices before they can exchange information. To avoid such inconvenience, a wireless communication technology has highly been demanded by users. Bluetooth is one of such technologies and especially attractive in the environment of hand-held devices since a low cost is an important requirement of such a technology [1]. Due to its low cost, Bluetooth is expected to be embedded in many consumer devices in the near future. Combined with the demand for access to the Internet everywhere in public places, a Bluetooth-based public Internet access point is expected to be popular.

While protocol models for Bluetooth-based access networks have been proposed in the literature [2][3], their performance is yet to be known. In this paper, we present a simulation model of Internet access based on the Bluetooth technology. The model consists of an Internet access point and a number of user devices each equipped with a Bluetooth radio. To evaluate the performance of the model, we have developed a simulator that accurately models the behavior of each user's requests to Internet access and actions occurred in the Internet access point to service the requests. Since the radio in the Internet access point is shared by multiple users, a suitable scheduling policy is necessary to select which user will be serviced next. A plain round-robin policy seems to be a good choice since it is simple to implement yet provides fairness. We also consider two other scheduling policies called a *weighted round-robin* and a *multi-level round-robin*. The simulation results show that both policies improve the performance by up to 35% over the plain round-robin policy.

Performance can further be improved by facilitating multiple Bluetooth radios in the Internet access point. It would be ideal if the performance boosts up to N times when N Bluetooth radios are used. However, since more than one radio may use the same carrier band due to the nature of Frequency Hopping Spread Spectrum (FHSS) [4], the effective performance is less than N times especially when N becomes larger. This paper analyzes the performance of a Bluetooth system with multiple radios using a probabilistic model reflecting the interference.

This paper is organized as follows. The following section describes the Internet access model and the simulation methodology. Section III presents the simulation results along with detailed analysis. Section IV analyzes the impact of multiple Bluetooth radio units in the Internet access point. Finally, Section V gives concluding remarks.

II. Evaluation Model and Methodology

A Bluetooth-based Internet access point is a system that provides Bluetooth-equipped devices with access to the Internet. It consists of a Bluetooth radio unit through which data communications occur with user devices, and an interface to the wire-based network infrastructure leading to the Internet. The main role of the Internet access point is to establish a wireless connection to each user device and forward user's packets to/from the wired network.

A Bluetooth-based Internet access point may have either

This research was supported in part by the Ministry of Education under the BK21 program. The first author was supported from the Basic Research Program (for the woman scientists) of the Korea Science & Engineering Foundation.

0-7695-0951-7/01 $10.00 © 2001 IEEE

a single or multiple radio units. Also, the radio units may act as a master or a slave. This gives us a design space encompassing four possible combinations: (1) single slave unit, (2) single master unit, (3) multiple slave units, (4) multiple master units. In the first combination, the Internet access point acts as a slave while each user device acts as a master establishing it's own piconet to connect to the Internet access point. In this setup, the total bandwidth of the radio link is shared by all of the user devices. However, this design requires the slave unit to switch from one piconet to another, which takes up to one frame. This overhead is significant when switching occurs frequently, e.g., in the order of a few frames.

The second design choice is the Internet access point with a single radio unit that acts as a master of a piconet with all user devices as slaves. When the radio is in the master mode, it controls all the slaves participating in the piconet by determining which slave exchanges data with the master. This design is more efficient than the first design since the overhead of piconet switching is not present.

The remaining two designs are extensions of the two initial designs with additional radio units in the Internet access point. This allows load-sharing between radio units and hence improves both the total bandwidth and the maximum number of users, at the costs of additional radio units. However, multiple radio units may interfere each other by occasionally colliding at the same frequency band, whose performance impact is analyzed in Section IV.

In the simulation, we assume the second design. We also assume that the number of slaves is limited to seven in order to remove the effect of parking that is required when more than seven slaves are to be connected to a master. We focus on the throughput and delay per user in a steady-state environment where no users join or depart the piconet. For the evaluation, we have developed an event-driven, packet-level simulator using PARSEC developed at UCLA as a successor to Maisie [5].

As for the Internet access model, we use a closed queuing simulation model that processes Internet web surfing traffic as shown in Figure 1. In this model, a user with a notebook generates traffic by clicking on the screen after consuming a *UserThinkTime*. The user's request is inserted into a upload queue in which the transmission is delayed until the master's polling is received. This delay is called a *Bluetooth upload scheduling delay* and dependent on the scheduling policy of the master. After the *upload scheduling delay*, the notebook that is polled by the master sends the user's request to the Internet Access Point(AP) and the AP forwards the request to the Internet. *InternetAccessTime* is the time for the AP to receive the user-requested data from the Internet since the request is submitted. When the requested data is ready, the data is inserted into the download queue and stays there until the corresponding notebook is polled. This delay is called a *Bluetooth download scheduling delay* and also affected by the scheduling policy of the master. At this time, AP performs segmentation if necessary based on the length of the message.

The packets used in the experiments include NULL and

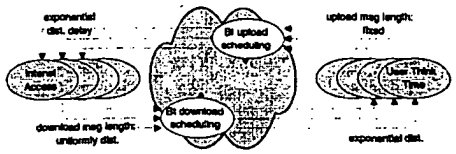


Fig. 1. Bluetooth simulation model

TABLE I
PARAMETER VALUES FOR EXPERIMENT

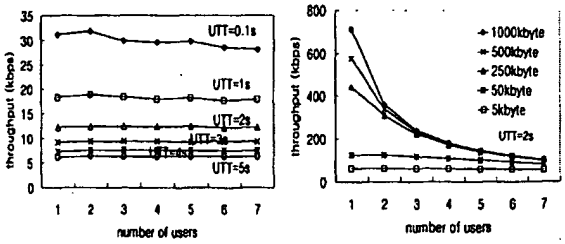
Parameter	Average Values
Message Length	[5 Kbytes, 50 Kbytes, 250 KBytes 500 Kbytes, 1000 Kbytes]
InternetAccessTime	[0.5s, 1s]
UserThinkTime	[0.1s, 1s, 2s, 3s, 4s, 5s]

POLL packets for control data transmission, and DH1, DH3, and DH5 packets for user data transmission. We assume fixed-length upload data and variable-length download data reflecting the asymmetric nature of the Internet traffic. The parameter values used in the simulation are shown in Table I. The *InternetAccessTime* and the *UserThinkTime* are random variables distributed exponentially with averages shown in the table. The message length is also a random variable, distributed uniformly. Each simulation is performed for a window of 24 hours.

We measure the performance by throughput and delay. The throughput is defined as the rate of data transmitted to a slave in a time unit. As for the delay, two measurements are considered: *completion delay* and *user delay*. The user delay is defined as the time between the user's click and the arrival of the first packet of the requested data. It is equal to the sum of *Bluetooth upload scheduling delay*, *InternetAccessTime*, and *download scheduling delay*. The completion delay is defined as the time between the user's click and the arrival of the last packet of the requested data. The completion delay represents the time until the requested service is completed.

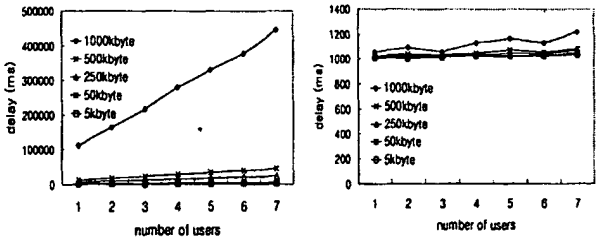
As mentioned before, the Bluetooth upload and download scheduling delays are affected by the scheduling policy of the master. In Bluetooth specification 1.0, a round-robin (RR) is implicitly assumed as the underlying scheduling policy. In an RR policy, a master polls its slaves one by one in a fixed order. A slave is allowed to transmit a message in a designated slot only when it has been addressed by the master in the preceding slot. Thus the polling sequence of the master is critical to the Bluetooth scheduling delay. The RR policy is fair in the sense that each user receives the same number of polls. Although it seems fair, a slave with download data should wait for other slaves even when the other slaves do not need to be serviced.

In this paper, we consider two other scheduling policies to improve the delay: *weighted round robin* (weighted RR) and *multi-level round robin* (multi-level RR). In the weighted RR, the master polls a slave up to n times successively when there is data that needs to be transmitted to the slave, while slaves with no data is polled only once.



(a) Message length = 5 Kbytes (b) Various message length

Fig. 2. Throughput per user



(a) Completion delay (b) User delay

Fig. 3. The variation of delay for the message length

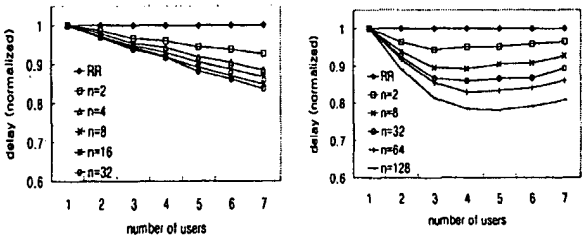
This policy improves the completion delay as we will see in the next section. On the other hand, the multi-level RR can improve the user delay, as well as the completion delay by grouping the slaves into two classes, one consisting of slaves with download data and the other consisting of slaves with no data. In this policy, the master polls all the slaves belonging to the first class up to n times successively, prior to polling the slaves belonging to the second class, which are polled only once as in the case of the weighted RR.

III. Performance Evaluation

This section presents the simulation results based on the Internet access model described in the previous section. Figure 2-a shows the throughput per user for various numbers of users and *UserThinkTimes* (denoted as *UTT*) in the Internet access model where the average message length is 5 Kbytes and *InternetAccessTime* is 1 second in average. As we can see from the figure, the throughput is largely independent on the number of users when the piconet is not overloaded. Figure 2-b shows simulation results where the average message length is varied from 5 Kbytes to 1000 Kbytes while *UserThinkTimes* is fixed to 2 seconds. In contrast to the previous results, when the average message length exceeds 250 Kbytes, throughput per user decreases by up to the ratio of $1/n$ as the number of users increases to n. However, it still outperforms the fastest dial-up modem (56 Kbps) in the marketplace even when a single Internet access point is shared by seven users.

Figure 3 shows the performance in terms of delay. Figure 3-a shows the completion delay for various message lengths ranging from 5 Kbytes to 1000 Kbytes. The completion delay increases up to 50 seconds as the number of users increases when the average message length is 1000 Kbytes due to the heavy traffic. However, when the average message length is below 5 Kbytes, the completion delay is less than 5 seconds regardless of the number of users. Figure 3-b shows the measurement of user delay in the same environment. In contrast to the completion delay, variation of user delay is very small for the range of message lengths we consider. Moreover, most of the user delay is due to the *InternetAccessTime* which is fixed to 1 second in the simulation. The above results indicate that the Bluetooth-based Internet access service is feasible in terms of delay as well as throughput.

In the previous experiments, it is assumed that the



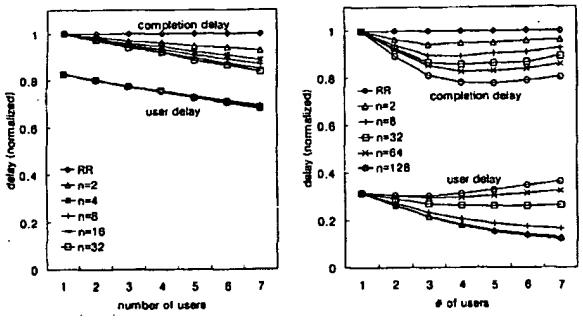
(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

Fig. 4. Completion delay in weighted RR

round-robin is the underlying scheduling policy. In the following, we present the simulation results for the other two policies relative to the round-robin assuming *InternetAccessTime* and *UserThinkTime* are 0.5 sec and 1 sec in average, respectively. Figure 4 shows the completion delay of weighted RR normalized to RR for various numbers of successive polls for each user, denoted as n. The results show that the weighted RR policy improves the completion delay by up to 15% when the traffic is low, as shown in Figure 4-a. When the traffic is heavy, the improvement is even more evident as shown in Figure 4-b, where improvement is by up to 20%. Similar results are observed for other values of parameters, although they are not included in this paper.

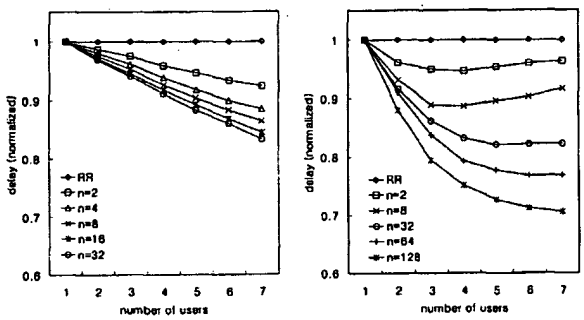
However, weighted RR increases the user delay although it improves the completion delay. Figure 5 shows the measurement of the user delay as well as the completion delay, both of which normalized as the completion delay of RR. When the load is low, increase in the user delay is marginal as shown in Figure 5-a. However, when the piconet is overloaded, the user delay increases by up to 24% compared to RR as shown in Figure 5-b. Note that the increase of the user delay is relatively small compared to the reduction in the completion delay. Overall, weighted RR policy gives performance improvement of up to 20% compared to RR.

The delay of weighted RR can be improved by giving priority to the users with download data as explained in Section II. Figure 6 shows the completion delay of the multi-level RR normalized to the delay of RR. The figure shows that weighted RR improves the completion delay by up to 25% when the load is high, and by up to 15% even when the load is low. From the results, we can notice that multi-level RR is even more efficient than weighted RR



(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

Fig. 5. User delay in weighted RR



(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

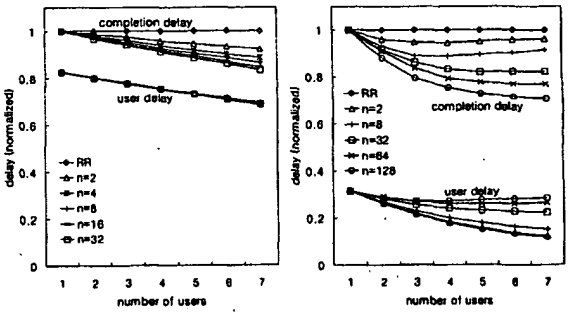
Fig. 6. Completion delay in multi-level RR

especially in an overloaded situation. The results also show that the best performance is achieved when the parameter value of n is set properly based on the transaction unit.

Figure 7 shows the results of the user delay along with the completion delay. Similarly to weighted RR, when the load is low, difference in the user delay is marginal. However, when the load is high, the user delay increases by up to 15%. Note that this number is much smaller than the case of weighted RR, where the increase is by up to 24%. The results show that the multi-level RR improves the completion delay better than weighted RR with smaller increase in the user delay. Thus, in the Internet web surfing environment with burst traffic, scheduling policies based on a transaction unit such as the weighted RR and the multi-level RR are better than policies based on a frame unit such as the pure round robin.

IV. Analysis on Inter-channel Interference

In the previous sections, a single Bluetooth unit in the Internet access point is assumed to service several users by forming a piconet consisting of all the users and the Internet access point. In this setting, all piconet members share a single Bluetooth channel under the control of the master of the piconet. A Bluetooth channel follows a unique frequency hopping sequence determined by the Bluetooth address of the master. Since different Bluetooth channels use different frequency hopping sequences, more than one



(a) Message length = 5 Kbytes (b) Message length = 50 Kbytes

Fig. 7. User delay in multi-level RR

Bluetooth channel can co-exist in a single Bluetooth cell. This gives a design choice of facilitating multiple Bluetooth units in a single Internet access point. However, multiple channels interfere with each other by occasionally using the same frequency band. Specifically, there are 79 different frequency bands defined in the Bluetooth specification [6] and the probability of two independent channels using the same frequency band at any given time is 1/79. If both channels transmit a message simultaneously when this happens, a collision occurs and eventually the message is garbled.

Of course, Bluetooth has a mechanism to detect if a message is garbled and retransmit it if so. On the receipt of a message, the recipient sends the acknowledgement to the sender, usually piggybacking on the next message. If the sender does not receive a proper acknowledgement, the message is retransmitted, usually at the next hop. Note that Bluetooth does not require a back-off mechanism to avoid repetitive collisions on retransmission, in contrast to other MAC protocols such as ALOHA since the probability of collision at the next hop is independent of the collision. This also simplifies the probabilistic model of collisions as given in the following.

We begin our analysis with the probability of a message transmitted in a channel being garbled by another channel. For simplicity, we assume that each message occupies a single Bluetooth slot and all channels are aligned each other so that the start of each slot is synchronized. Under this assumption, the probability of a message being garbled by another channel is the product of two probabilities, the probability of two channels hopping onto the same frequency band and the probability of a message being carried in a slot. The latter can be thought of as a normalized load carried over a channel including both new messages and retransmitted messages. If we denote this term as G, the probability of a message being garbled is given as $G/79$. Then the probability of a message not being garbled when there are N channels with the same characteristic, is equal to the probability of a message not garbled by any of $N-1$ channels. Hence, we get

$$P_{nocollision}(N) = (1 - G/79)^{N-1}. \quad (1)$$

In addition to the message, the corresponding acknowl-

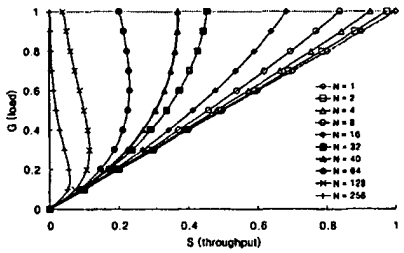
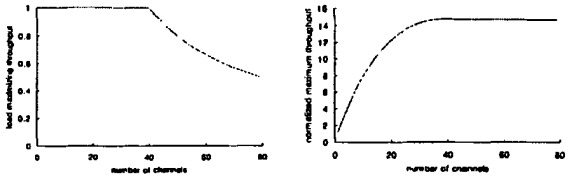


Fig. 8. Throughput characteristic



(a) Carried load (b) Maximum throughput

Fig. 9. Maximum throughput analysis

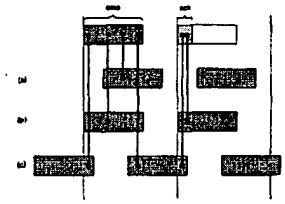


Fig. 10. Alignment of channels

edgement also needs to be transmitted without being garbled in order that a message is regarded as transmitted successfully by the sender. If we ignore the case of possible forward error correction and assume that the probability of an acknowledgement being garbled is the same as a message being garbled, the probability of a transmission being successful is:

$$P(N) = P_{nocollision}(N)^2 = (1 - G/79)^{2(N-1)} \quad (2)$$

Given the probability of a transmission being successful $P(N)$ and the carried load G, we obtain the ratio of messages transmitted successfully on a channel, i.e., *throughput*, by multiplying G by $P(N)$.

$$S = GP(N) = G(1 - G/79)^{2(N-1)} \quad (3)$$

The aggregated throughput of all of N channels, i.e., the *throughput of the Internet access point*, is given simply as S multiplied by N:

$$S(N) = NGP(N) = NG(1 - G/79)^{2(N-1)} \quad (4)$$

The relationship between S and G for various N is shown in Figure 8. The figure shows that for smaller N the throughput S increases almost linearly as the carried load G increases. However, for large N, throughput does not increase beyond certain points, similarly to ALOHA [7]. For example, when N is 128, S increases towards the maximum value until G reaches about 0.3 and drops beyond this point. Such threshold can be obtained by differentiating S by G.

$$\begin{aligned} \frac{dS}{dG} &= \frac{d}{dG}G(1 - G/79)^{2(N-1)} \\ &= (1 - G/79)^{2(N-1)-1}(1 - \tfrac{2N-1}{79}G) \end{aligned} \quad (5)$$

By setting Equation 5 equal to zero, we get $G = 79/(2N - 1)$. Since $G \leq 1$ by definition, we get $N > 40$. This implies that such threshold exists only when N is greater than 40. Thus the threshold G_{max} is represented as follows:

$$G_{max} = \begin{cases} 1 & \text{if } N \leq 40 \\ \frac{79}{2N-1} & \text{otherwise} \end{cases} \quad (6)$$

G_{max} for ranges of N is plotted in Figure 9-a. Replacing G in Equation 6 with G_{max}, we obtain the maximum throughput $S_{max}(N)$ as shown in Figure 9-b. The figure clearly shows that the maximum throughput does not improve when N increase beyond 40. From the results, we can conclude that the performance of the Internet access point improves as the number of Bluetooth units increases up to 40 although the improvement becomes smaller due to inter-channel interference.

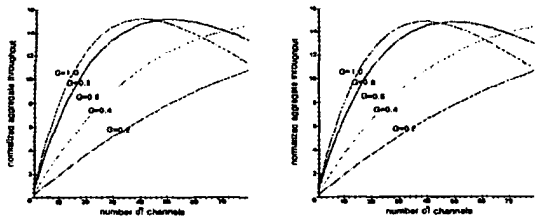
The previous analysis assumes that each message occupies a single slot and the channels are aligned with each other as shown in Figure 10-b (Case B). In this subsection, we try to relax the second assumption. When channels are not aligned as in the case shown in Figure 10-c (Case C), a message can collide by either of two messages in a channel, increasing the probability of a collision. On the other hand, Figure 10-a shows the case where the acknowledgement cannot be garbled by the other channel, improving the probability of a transmission being successful. Assuming that the three cases are generated randomly and that all the messages are carried over DH1 packets, the probability of each case is given as follows:

$$\begin{aligned} p_1 &= \frac{l-m-h}{l} = \frac{133}{625} = 0.2128, \\ p_2 &= \frac{l-m+h}{l} = \frac{385}{625} = 0.6160, \\ p_3 &= \frac{2m-l}{l} = \frac{107}{625} = 0.1712 \end{aligned} \quad (7)$$

where p_1, p_2, and p_3 represent the probability of two channels being in Case A, Case B, and Case C, respectively, and h, m, and l represent the length of the header of a DH1 packet (126 μsec), the length of a DH1 packet including the header (366 μsec), and the length of a slot (625 μsec), respectively.

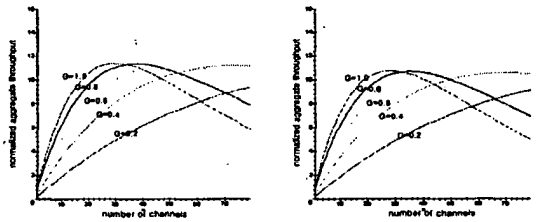
Applying binomial twice, we obtain the normalized throughput $S'(N)$ as follows:

$$\begin{aligned} S'(N) &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{N-1}{i}\binom{N-1-i}{j} \\ &\quad \times p_1^{N-1-(i+j)}NG(1 - G/79)^{N-1-(i+j)} \\ &\quad \times p_2^i NG(1 - G/79)^{2i} \\ &\quad \times p_3^j NG(1 - G/79)^{3j} \\ &= NG(1 - G/79)^{N-1} \\ &\quad \times \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\binom{N-1}{i}\binom{N-1-i}{j} \\ &\quad \times p_1^{N-1-(i+j)}p_2^i p_3^j(1 - G/79)^{i+2j} \end{aligned} \quad (8)$$



(a) aligned-channel model (b) unaligned-channel model

Fig. 11. Comparison of both models (DH1)



(a) DH3 (b) DH5

Fig. 12. Throughput in on unaligned channels

Figure 11 compares the previous model (*aligned-channel model*) and the relaxed model (*unaligned-channel model*). The figure shows that the difference between two models is marginal. It is due to the fact that the Case A (probability: 0.2128) and Case C (probability: 0.1712) offset each other and Case B (probability: 0.6160) dominates, which is the same case as the previous model.

In the case of DH3 or DH5, the equation remains unchanged but the probability p_1, p_2, and p_3 are computed in a slightly different manner:

$$\begin{aligned}
p_1^{DH3} &= \frac{3l-m^{DH3}-h}{3l} = \frac{151}{1875} = 0.0806, \\
p_2^{DH3} &= \frac{3l-m^{DH3}+h}{3l} = \frac{403}{1875} = 0.2149, \\
p_3^{DH3} &= \frac{2m^{DH3}-3l}{3l} = \frac{1321}{1875} = 0.7045, \\
p_1^{DH5} &= \frac{5l-m^{DH5}-h}{5l} = \frac{137}{3125} = 0.0438, \\
p_2^{DH5} &= \frac{5l-m^{DH5}+h}{5l} = \frac{389}{3125} = 0.1565, \\
p_3^{DH5} &= \frac{2m^{DH5}-5l}{5l} = \frac{2599}{3125} = 0.8317
\end{aligned} \quad (9)$$

where m^{DH3} and m^{DH5} are the size of a DH3 packet (1598 μsec) and the size of a DH5 packet (2862 μsec), respectively. The results are shown in Figure 12. Since p_3 is greater compared to the case of DH1, the normalized throughput is reduced due to a larger collision probability. Note that this does not imply that the performance is reduced by using DH3 / DH5 instead of DH1, since the throughput shown in the figure is normalized by the throughput of a single channel using DH3 and DH5, respectively. In fact, the use of DH3 / DH5 improves the performance since the overhead of packetization is much less than DH1. From the results, we can notice that both models show similar characteristics and reflect the relationship between the number of channels and the resulting aggregated throughput.

V. Conclusion

Today, access to the Internet is essential in every-day life. It is hard to think of business, research, and entertainment without the Internet. Therefore, it is a requirement, rather than an option, to facilitate functionality of Internet access in hand-held devices such as PDA for *everywhere-Internet* services. Since hand-held devices require low cost and low power consumption, Bluetooth has been regarded as one of the most promising solutions for a wireless connection between the user devices and the wired network infrastructure.

In this paper, we have presented preliminary results on the performance of a Bluetooth-based access network. In the performance evaluation, we have modeled a simple Internet access scenario and simulated while varying the values of various parameters including user's think times, message lengths, and Internet access delay. The simulation results indicate that the Bluetooth-based access network provides performance of tens of Kbps to hundreds of Kbps depending on the number of users. This result is encouraging since the Bluetooth-based access network gives performance significantly better than the traditional dial-up network without the tedious wire connection. The performance can be improved if multiple Bluetooth radio units are used in the Internet access point. We have shown that the performance improves as the number of Bluetooth units increases up to 40 based on an analytical model.

As a future work, we plan to develop a more general Bluetooth simulation package including a traffic generator that reflects a variety of traffic such as ftp, e-mail, and name-card exchange. We expect the simulation package will allow us to evaluate various aspects of Bluetooth systems including issues on scheduling policies, segmentation and reassembly, parking of user devices that are largely ignored in this paper. We also expect thorough evaluation with the new simulation tool would reveal possible causes of performance bottlenecks in the Bluetooth system.

References

[1] J. Haartsen, M. Naghshineh, J. Inouye, O. J. Joeressen, and W. Allen, "Bluetooth: vision, goals, and architecture," *Mobile Computing and Communications Review*, vol. 2, pp. 38–45, Oct. 1998.

[2] P. Bhagwat, I. Korpeoglu, C. Bisdikian, M. Naghshineh, and S. K. Tripathi, "BlueSky: A cordless networking solution for palmtop computers," in *Proceedings of the 5th International Conference on Mobile Computing and Networking (ACM Mobicom '99)*, pp. 69–76, 1999.

[3] M. Albrecht, M. Frank, P. Martini, M. Schetelig, A. Vilavaara, and A. Wenzel, "IP services over Bluetooth: Leading the way to a new mobility," in *Proceedings of the 24th Conference on Local Computer Networks*, pp. 2–11, 1999.

[4] J. Haartsen, "The bluetooth radio system," *IEEE Personal Communications*, vol. 7, pp. 28–36, Feb. 2000.

[5] The Parallel Simulation Environment for Complex Systems, "http://pcl.cs.ucla.edu/projects/parsec/."

[6] The Bluetooth Special Interest Group, "http://www.bluetooth.com/techn/index.asp," Feb. 1999.

[7] M. Schwartz, *Computer-communication network design and analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[8] P. Johansson, N. Johansson, U. Körner, J. Elg, and G. Svennarp, "Short range radio based ad-hoc networking: performance and properties," in *Proceedings of the IEEE International Conference on Communications (ICC '99)*, vol. 3, pp. 1414–1420, 1999.

Performance evaluation of the Bluetooth-based public Int point

Yujin Lim Jesung Kim Sang Lyul Min Joong Soo Ma
Dept. of Mech. Eng., Seoul Univ., South Korea;

This paper appears in: **Information Networking, 2001. Proceedings. 15th Internationa**
Publication Date: 31 Jan.-2 Feb. 2001
On page(s): 643 - 648
Number of Pages: xxiii+942
Meeting Date: 01/31/2001 - 02/02/2001
Location: Beppu City, Oita
ISBN: 0-7695-0951-7
INSPEC Accession Number:7017461
Digital Object Identifier: 10.1109/ICOIN.2001.905527
Posted online: 2002-08-07 00:18:23.0

Abstract
Bluetooth has been regarded as a promising solution to an inexpensive wireless connec application of **Bluetooth** technology has been focused mainly on replacing cables betwe general wireless telecommunication such as public Internet access via a **Bluetooth**-equi expected to be one of the most popular applications in the near future. However, it is not whether the performance of **Bluetooth**-based systems is sufficient for such an applicatio preliminary results of performance evaluation of a **Bluetooth**-based Internet access poin based on simulation of an Internet access model consisting of a **Bluetooth**-based netwo number of **Bluetooth**-equipped notebook computers. The simulation results indicate that performance comparable to the fastest dial-up modem even when a number of users sha radio unit. Better performance is expected when more than one **Bluetooth** radio unit are each unit services different users concurrently. However, **Bluetooth** units in a single radi interferes with each other since the channels established by each unit occasionally collid frequency band. This paper analyzes the impact of such interference based on an analyti channel interference. The analysis shows that the performance improves as the number up to 40. We expect the proposed inter-channel interference model is useful in the desig facilitating multiple **Bluetooth** units

Index Terms
Inspec
Controlled Indexing
Internet adjacent channel interference land mobile radio notebook computers performance evaluation radio access networks

Non-controlled Indexing
Bluetooth radio unit **Bluetooth**-based public Internet access point **Bluetooth** notebook computers PDA RFI analytical model celluar phones dial-up mod frequency band hand-held devices inexpensive wireless connection intercha interference model performance evaluation simulation results

Author Keywords
Not Available

References

No references available on IEEE Xplore.

Citing Documents

1 An improved packet collision analysis for multi-Bluetooth piconets considering freque time effect, Ting-Yu Lin; Yen-Ku Liu; Yu-Chee Tseng
Selected Areas in Communications, IEEE Journal on
On page(s): 2087- 2094, Volume: 22, Issue: 10, Dec. 2004
Abstract | Full Text: PDF (672)

2 Performance evaluation in Bluetooth dense piconet areas, Mazzenga, F.; Cassioli, D. Loreti, P.; Vatalaro, F.
Wireless Communications, IEEE Transactions on
On page(s): 2362- 2373, Volume: 3, Issue: 6, Nov. 2004
Abstract | Full Text: PDF (784)

View Search Results | Next Article

Bluetooth: Technology for Short-Range Wireless Apps

Pravin Bhagwat • *Reefedge Inc.*

Handheld devices are rapidly becoming an integral part of our daily lives, and many road warriors already carry a cell phone, palmtop, and laptop computer with them. In most cases, these devices do not have compatible data communication interfaces, or, if they do, the interface requires cumbersome cable connections and configuration procedures. An obvious solution is to get rid of the cables and use short-range wireless links to facilitate on-demand connectivity among devices. An ideal solution would also be inexpensive, enabling of compelling applications, and universally adopted by device vendors.

In 1998, five major companies (Ericsson, Nokia, IBM, Toshiba, and Intel) formed a group to create a license-free technology for universal wireless connectivity in the handheld market. The result is Bluetooth, a technology named after a 10th-century king who brought warring Viking tribes under a common rule. The Bluetooth specifications,[1,2] currently in version 1.1, define a radio frequency (RF) wireless communication interface and the associated set of communication protocols and usage profiles.

The link speed, communication range, and transmit power level for Bluetooth were chosen to support low-cost, power-efficient, single-chip implementations of the current technology. In fact, Bluetooth is the first attempt at making a single-chip radio that can operate in the 2.4-GHz ISM (industrial, scientific, and medical) RF band. While most early Bluetooth solutions are dual chip, vendors have recently announced single-chip versions as well. In this overview of the technology, I will first describe the lower layers of the Bluetooth protocol stack. I will also briefly describe its service discovery protocol and, finally, how the layers of the protocol stack fit together from an application's point of view.

Bluetooth Specifications

The Bluetooth 1.1 specification was released in February 2001. The specification consists of two parts: core and profiles.

Core Specifications

The core specification defines all layers of the Bluetooth protocol stack.[1] As shown in Figure 1, the Bluetooth stack differs from the classical seven-layer networking model in some ways. These differences are primarily to support ad hoc connectivity among participating nodes, while conserving power and accommodating devices that lack resources to support all layers of the classical networking stack.

The radio is the lowest layer. Its interface specification defines the characteristics of the radio front end, frequency bands, channel arrangements, permissible transmit power levels, and receiver sensitivity level. The next layer is the baseband, which carries out Bluetooth's physical (PHY) and media access control (MAC) processing. This includes tasks such as device discovery, link formation, and synchronous and asynchronous communication with peers. Bluetooth peers must exchange several control messages for the purpose of configuring and managing the baseband connections. These message definitions are part of the link manager protocol (LMP). The functional entity responsible for carrying out the processing associated with LMP is called the *link manager.*

Bluetooth is unique in offering the front-end RF processing integrated with the baseband module. On-chip integration lowers the cost of the network

 1089-7801/01/$10.00©2001 IEEE

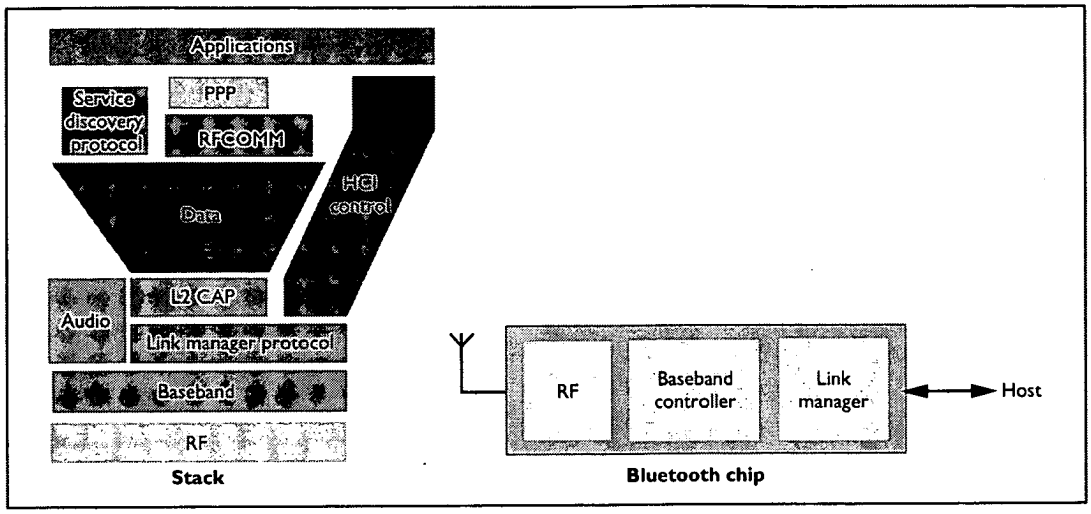


Figure 1. The Bluetooth networking stack and chip. The design supports the integration of an analog radio front end, signal-processing elements, and baseband controller on a single chip.

interface, and the small size makes it easy to embed Bluetooth chips in devices such as cell phones and PDAs. A Bluetooth chip can be connected to its host processor using USB, UART, or PC-card interfaces.

The Host Controller Interface (HCI) specification defines a standard interface-independent method of communicating with the Bluetooth chip. The software stack on the host processor communicates with the Bluetooth hardware using HCI commands. Since no hardware-specific knowledge is needed, the Bluetooth stack software can easily be ported from one Bluetooth chip to another. The HCI layer is part of the Bluetooth stack, but it doesn't constitute a peer-to-peer communication layer since the HCI command and response messages do not flow over the air link.

The logical link control and adaptation protocol (L2CAP) specification can be viewed as Bluetooth's link layer. Usually, L2CAP and layers above it are implemented in software. L2CAP delivers packets received from higher layers to the other end of the link. Bluetooth devices can establish an L2CAP connection as soon as they are in range of each other. A client device then needs to discover the services provided by the server device.

The service discovery protocol (SDP) defines the means by which the client device can discover services as well as their attributes. The SDP design has been optimized for Bluetooth. It defines only the discovery mechanisms; the methods for accessing those services are outside its scope.

The RFCOMM specification defines a method of emulating the RS-232 cable connection on top of the Bluetooth airlink. RFCOMM supports legacy applications that use the COM port to communicate with the peer host. For example, point-to-point (PPP) protocols expect a serial line interface from the lower layer. Since PPP provides a packet-oriented interface to the higher layers, all packet-based network and transport protocols, including TCP/IP, can be supported on top of PPP. More efficient methods of running IP over Bluetooth are currently under development.

Profile Specifications

Vendors can use the services offered by the Bluetooth stack to create a variety of applications. Because interoperability is crucial to Bluetooth's operation, the Bluetooth SIG has defined profile specifications to support it.[2] The current specifications include 13 profiles listed in Table 1 (next page).

The profiles specify controller and stack parameter settings as well as the features and procedures required for interworking among Bluetooth devices. All vendor implementations of these profiles are expected to be interoperable. The Bluetooth certification authority uses the profiles to test and certify compliance, and grants use of the Bluetooth logo only to products that conform to the methods and procedures defined in the profiles.

Radio Front End

The 2.4-GHz ISM band in which Bluetooth operates is globally available for license-free use. Europe and the United States allocate 83.5 MHz to this band, but Spain, France, and Japan allocate less. To accommodate these differences, 79 channels spaced 1 MHz apart are defined for Europe and the U.S., and 23 RF channels spaced

Table 1. Profiles defined in Bluetooth 1.1 specifications.

Use case	Description
Generic access	Generic procedures for discovery and link management of connecting to Bluetooth devices.
Service delivery	Features and procedures for a Bluetooth device application to discover services registered in other devices.
Cordless telephone	Features and procedures for interoperability between different units active in a "3-in-1"phone.
Intercom	Requirements for supporting intercom functionality within a "3-in-1" phone.
Serial port	Requirements for setting up emulated serial cable connections using RFCOMM between two peer devices.
Headset	End-user service requirements and interoperability features for Bluetooth devices implementing headsets.
Dial-up networking	End-user service requirements and interoperability features for Bluetooth devices implementing dial-up networking.
Fax	End-user service requirements and interoperability features for Bluetooth devices implementing fax services.
LAN access	Definition of (a) how Bluetooth devices can access LAN services using PPP and (b) how the PPP mechanisms form a network.
Generic object exchange	Requirements for Bluetooth devices to support object exchange usage models.
Object push	Application requirements for Bluetooth devices to support the object push usage model.
File transfer	Application requirements for Bluetooth devices to support the file transfer usage model.
Synchronization	Application requirements for Bluetooth devices to support the synchronization usage model.

1 MHz apart are defined for Spain, France, and Japan. Efforts are under way to open up the full width of the spectrum in Spain and France, as well as in Japan so that Bluetooth devices would function worldwide.

Bluetooth is a frequency-hopping spread-spectrum system. This means that the radio hops through the full spectrum of 79 or 23 RF channels using a pseudorandom hopping sequence. The hopping rate of 1,600 hops per second provides good immunity against other sources of interference in the 2.4-GHz band. The link speed is 1 Mbps, which is easily achieved using a simple modulation technique (Gaussian Frequency Shift Keying, or GFSK). A more complex modulation technique could achieve a higher rate, but GFSK keeps the radio design simple and low cost.

The radio front end is usually the most costly part of a wireless network interface. In typical radio receivers, the RF filters, oscillators, and image-reject mixers process input signals at high frequencies. Such circuits require expensive materials. To keep costs down, Bluetooth recommends shifting the input signal to a lower intermediate frequency (IF, around 3 MHz), which allows on-chip construction of low-power filters using CMOS material. Shifting to low IF, however, creates new problems, such as reduced receiver sensitivity. Recommended receiver sensitivity for Bluetooth is −70 dBm or better. The comparable number for IEEE 802.11 Wireless LANs is about −90 dBm). Thus, for the same transmit power, the range for Bluetooth is shorter than it is for 802.11 WLAN.

Piconets and Scatternets

A set of Bluetooth devices sharing a common channel is called a *piconet*. As shown on the left side of Figure 2, a piconet is a star-shaped configuration in which the device at the center performs the role of *master* and all other devices operate as *slaves*. Up to seven slaves can be active and served simultaneously by the master. If the master needs to communicate with more than seven devices, it can do so by first instructing active slave devices to switch to low-power park mode and then inviting other parked slaves to become active in the piconet. This juggling act can be repeated, which allows a master to serve a large number of slaves.

Most envisioned Bluetooth applications involve local communication among small groups of devices. A piconet configuration consisting of two, three, or up to eight devices is ideally suited to meet the communication needs of such applications. When many groups of devices need to be active simultaneously, each group can form a separate piconet. The slave nodes in each piconet stay synchronized with the master clock and hop according to a channel-hopping sequence that is a function of the master's node address. Since channel-hopping sequences are pseudorandom, the probability of collision among piconets is small. Piconets with overlapping coverage can coexist and operate independently. Nonetheless, when the degree of overlap is high, the performance of each piconet starts to degrade.

In some usage scenarios, however, devices in different piconets may need to communicate with

each other. Bluetooth defines a structure called *scatternet* to facilitate interpiconet communication. A scatternet is formed by interconnecting multiple piconets. As shown on the right side of Figure 2, the connections are formed by *bridge nodes*, which are members of two or more piconets. A bridge node participates in each member piconet on a time-sharing basis. After staying in a piconet for some time, the bridge can turn to another piconet by switching to its hopping sequence. By cycling through all member piconets, the bridge node can send and receive packets in each piconet and also forward packets from one piconet to another.

A bridge node can be a slave in both piconets or be a slave in one and a master in another.[3] For example, consider a room full of people, where each person has a cell phone and a cordless headset. When users speak into their headsets, only the cell phones paired with their headsets should pick up the signal. In this example, each headset and cell phone pair constitutes a separate piconet. Now suppose these users also want to send text messages from their cell phones to one another. This will be possible only if all piconets are interconnected to form a large scatternet.

The techniques for forming scatternets are still under development.[4]

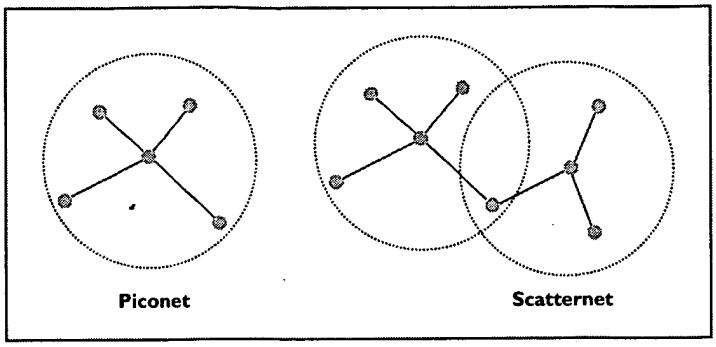


Figure 2. Piconet (left) and scatternet (right). The master device at the center of a piconet can serve up to seven slaves; members of two or more piconets are called bridge nodes, which support interpiconet communication.

Inquiry and Paging

Bluetooth uses a procedure known as *inquiry* for discovering other devices; it uses *paging* to subsequently establish connections with them. Both inquiry and paging are asymmetric procedures. In other words, they involve the inquirer and the inquired (as well as the pager or the paged) devices to perform different actions. This implies that when two nodes set up a connection, each needs to start from a different initial state; otherwise, they would never discover each other. The profile specifications play an important role here, defining the required initial state for each device in all usage scenarios. A symmetric procedure for establishing connections is an ongoing topic of research.[4]

The inquiry and paging are conceptually simple operations, but the frequency-hopping nature of the physical layer makes the low-level details quite complex. Two nodes cannot exchange messages until they agree to a common channel-hopping sequence as well as the correct phase within the chosen sequence. Bluetooth solves this problem simply by mandating the use of a specific inquiry-hopping sequence known to all devices. During inquiry, both nodes (one is the listener and the other is the sender) hop using the same sequence; but the sender hops faster than the listener, transmitting a signal on each channel and listening between transmissions for an answer. When more than one listener is present, their replies may collide. To avoid the collision, listeners defer their replies until expiration of a random backoff timer. Eventually the sender device collects some basic information from the listeners, such as the device address and the clock offsets. This information is subsequently used to page the selected listener device.

The communication steps during the paging procedure are similar, except that the paging message is unicast to a selected listener, so the listener need not back off before replying. The sender also has a better estimate of the listener's clock, which enables it to communicate with the listener almost instantaneously. Upon receiving an ACK for the paging message, the sender becomes the master and the listener becomes the slave of the newly formed piconet, and both nodes switch to the piconet's channel-hopping sequence. Later, if necessary, the master and slave roles can be swapped.

The steps for admitting a new slave into an existing piconet are slightly more complex. The master can either start discovering new nodes in its neighborhood and invite them to join the piconet or, instead, wait in scan (listen) state and be discovered by other nodes. With both options, communication in the original piconet must be suspended for the duration of the inquiry and paging process. The latency of admitting a new node into the piconet can be large if the master does not switch to the inquiry or scan modes frequently. This latency can be reduced only at the cost of some piconet capacity. The study of this trade-off is another topic of ongoing research.

Table 2. Channel throughput for different packet sizes.

Packet size (in slots)		Throughput in Kbps (with FEC)		Throughput in Kbps (no FEC)	
In slave direction	In master direction	In slave direction	In master direction	In slave direction	In master direction
1	1	108.8	108.8	172.8	172.8
3	1	387.2	54.4	585.6	86.4
5	1	477.8	36.3	723.2	57.6

Low-Power Modes

Bluetooth offers different low-power modes for improving battery life. Piconets are formed on demand when communication among devices is ready to take place. At all other times, devices can be either turned off or programmed to wake up periodically to send or receive inquiry messages. When a piconet is active, the slaves stay powered on to communicate with the master. It is possible to switch a slave into a low-power mode whereby it sleeps most of the time and wakes up only periodically.

Three types of low-power modes have been defined:

- *Hold mode* is used when a device should be put to sleep for a specified length of time. As described earlier, the master can put all its slaves in the hold mode to suspend activity in the current piconet while it searches for new members and invites them to join.
- *Sniff mode* is used to put a slave in a low-duty cycle mode, whereby it wakes up periodically to communicate with the master.
- *Park mode* is similar to the sniff mode, but it is used to stay synchronized with the master without being an active member of the piconet. The park mode enables the master to admit more than seven slaves in its piconet.

Piconet Channel

As soon as a piconet is formed, communication between the master and the slave nodes can begin. The piconet channel is divided into 625-microsecond intervals, called slots, where a different hop frequency is used for each slot. The channel is shared between the master and the slave nodes using a frequency-hop/time-division-duplex (FH/TDD) scheme whereby master-slave and slave-master communications take turns. Slave-to-slave communication is not supported at the piconet layer. If two slaves need to communicate peer to peer, they can either form a separate piconet or use a higher layer protocol, such as IP over PPP (see Figure 1), to relay the messages via the master.

At a 1-Mbps link speed, a 625-microsecond slot time is equivalent to the transmission time of 625 bits. However, a single slot packet size in Bluetooth is only 366 bits. This reserves enough guard time to let the frequency synthesizers hop to the next channel frequency and stabilize. Discounting space for the headers leaves 30 bytes for the user payload.

Synchronous Link

To transmit real-time voice, an application must reserve a slot in both directions at regular intervals. In Bluetooth terminology, this is called a synchronous (SCO) link. An SCO link can transport telephone-grade voice. The speech coder generates 10 bytes every 1.25 milliseconds. Since a baseband packet can carry up to 30 bytes in each slot, only one slot in each direction is needed every 3.75 ms (or every sixth slot). The packet type that carries 30 voice bytes is called an *HV3 packet*. This packet is transmitted without coding or protection, and is not retransmitted if it is lost.

To cope with bit errors when the channel conditions are not perfect, some forward error correction (FEC) should be added to the voice payload. An HV2 packet carries 20 bytes of voice plus 10 bytes of redundant data (2/3 FEC code). Since 20 bytes of speech is generated in 2.5 ms, the SCO link should reserve one slot in each direction every 2.5 ms (or every fourth slot). To cope with extreme channel conditions, the baseband specification also defines an HV1 packet that carries only 10 bytes of speech and 20 bytes of FEC code. An HV1 SCO link uses up the entire channel capacity. This means that all data transfer sessions will be suspended when an HV1 SCO connection is in progress.

Asynchronous Link

Data communication between a master-slave pair involves a different set of considerations. For example, the data payload must be protected by a cyclic redundancy check (CRC) so that the receiver can determine whether the received bits are in error. When losses occur, the baseband layer should retransmit the data. Furthermore, to make efficient use of the piconet channel, slots should be allocated on demand, instead of being reserved for the usage duration. A data path between a master-slave pair meeting all of these requirements is called an asynchronous data link (ACL). SCO links have pri-

ority over data, so ACLs can claim only unused slots. Only a single ACL can exist between a master and a slave.

The master is responsible for distributing available slots among all ACLs. This scheme has two advantages:

- the master can ensure that the slave transmissions do not collide; and
- the slots can be allocated to satisfy the quality of service (QoS) requirement of each ACL. The master can grant more bandwidth to a slave by polling it more frequently or by changing the packet size.

The baseband specification does not mandate the use of any specific slot-allocation scheme. Chip vendors can choose any policy that fits their target applications.

As with SCO packets, the payload size of single-slot ACL packets is limited to 30 bytes. After discounting space for the higher layer headers and the CRC, only 27 bytes are left to transport application data. When FEC is added, the available space goes down to 17 bytes. To improve channel efficiency, the baseband specification has defined multislot packets, which are three or five slots long and transmitted in consecutive slots. The transmitter stays fixed on a hop frequency during the length of packet transmission and skips over the missed hops after the transmission is complete. This reduces the effective channel-hopping rate, but increases the channel efficiency because of fewer hops.

Table 2 shows the achievable throughput in the master-to-slave and slave-to-master directions as a function of packet size, with and without FEC. Although link speed is 1 Mbps, achievable aggregate throughput can range from 217.6 Kbps to 780.8 Kbps. The presence of an HV3 or HV2 SCO link significantly reduces the achievable throughput of an ACL.

Logical Link Control and Adaptation Protocol

L2CAP can be viewed as the data plane of the Bluetooth link layer (see Figure 3). Because the baseband packet size is too small for transporting higher layer packets, a thin layer is needed for exporting a bigger packet size to the higher layers. While a number of generic segmentation and reassembly protocols could be used or adapted for use over ACLs, the Bluetooth SIG instead defined L2CAP, which is highly optimized to work in conjunction with the baseband layer. For example,

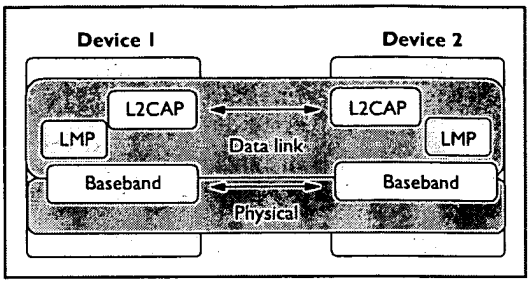


Figure 3. Lower part of stack. L2CAP can be viewed as the data plane of Bluetooth's link layer.

L2CAP does not support integrity checks because the baseband packets are already CRC protected. Likewise, it is assumed that the lower layer delivers packets both reliably and in sequence. These two assumptions significantly simplify the design of segmentation and reassembly logic. The only caveat is that L2CAP will not work if used over any media other than the Bluetooth baseband.

The multiplexing and demultiplexing of higher layer protocols is supported using channels, multiple instances of which can be created between any two L2CAP endpoints. Each higher layer protocol or data stream is carried in a different channel. The L2CAP channels are connection oriented in the sense that they require an explicit phase to establish the channel, during which both ends choose a local name (channel identifier) and communicate it to the other end. Subsequently, each packet sent over the channel is tagged with the channel identifier, which—within the context of the receiver—uniquely identifies the source as well as the protocol being transported over the channel.

The L2CAP specification also defines a connectionless channel for supporting broadcast and multicast group communication, but this feature is not yet fully developed.

Service Discovery Protocol

Both ends of a Bluetooth link must support compatible sets of protocols and applications to successfully exchange data. In some cases it may also be necessary to configure protocol and stack parameter settings before applications can be started. Such configuration settings cannot be chosen statically, since some parameters may require adjustment to match the features and services supported by the peer Bluetooth device.

Bluetooth's SDP provides a standard means for a Bluetooth device to query and discover services supported by a peer Bluetooth device. SDP is a client-server protocol. The server maintains a list of service records, which describe the characteris-

tics of services hosted at the server. By issuing SDP queries, a client can browse all available service records maintained at the server or retrieve specific attribute values from a service record.

In addition to defining query and response protocol formats, the SDP specification also defines a standard method for describing service attributes. Service attributes are represented using an <identifier, value> pair. The 1.1 specification defines some of the commonly used services, but developers have the freedom to define new subclasses of the standard services or to create new services on their own.

Since new service definitions do not require any coordination with the Bluetooth SIG numbering authority, it is necessary to ensure that two independently created service definitions do not conflict. Collisions are avoided by associating each service definition with a universally unique identifier (UUID) which is generated once at the time a service is defined. UUIDs of the services defined by the Bluetooth SIG are included in the assigned numbers document.

If the client already knows the UUID of the service it is looking for, it can query the SDP server for specific service attributes. Alternatively, the client can browse the list of available services and select from the list. These are the only two search options supported in SDP. Although other IP-based service discovery protocols, such as SLP and Jini, provide richer service description schema and more powerful search capabilities, the Bluetooth SDP has two advantages:

- The majority of version-1.1-compliant Bluetooth devices will be non-IP devices. Requiring them to support IP only for the sake of supporting SLP would be costly.
- SDP is optimized to run over L2CAP. Its limited search capabilities and non-text-based attribute-id and attribute-value descriptions lend an efficient and small footprint implementation for small devices.

SDP provides a mechanism only for retrieving service information from other devices. Methods of invoking those services are outside the scope of SDP.

Link Manager Protocol

Before a device can establish the L2CAP channel, the link manager must carry out a number of baseband-specific actions, such as piconet creation, master-slave role assignments, and link configuration. These functions belong to the control plane of the Bluetooth link layer and require the link manager to exchange LMP messages over the air link. Depending on the operating environment, the link manager must adjust a number of piconet and link-specific parameters. For example, the peer-link controller can be instructed to switch to a low-power mode, adjust its power level, increase the packet size, and change the requested QoS on an ACL.

Security can also be configured using LMP messages. Before a data or voice exchange can begin, Bluetooth devices should be able to authenticate each other. Likewise, transmission over the air link must be encrypted to provide protection from eavesdroppers. Both objectives are easy to achieve when a security association already exists between a pair of devices. The link manager can use the shared secret key to verify the peer device's authenticity as well as to negotiate a link key for encryption. A typical session between two Bluetooth devices begins with the formation of a piconet, followed by the exchange of LMP messages first to authenticate and then to negotiate new encryption keys with the peer device. Only upon successful completion of the LMP handshake can further data exchange or voice communication take place.

The level of security built into the version 1.1 specifications is satisfactory so long as the initial security associations are computed in a secure fashion. The baseband and LMP specifications also define a method, called *pairing*, for creating a new security association between two devices when they pair for the first time. The method uses an out-of-band channel for creating a security association, which is then used as a seed to compute a cryptographically secure shared secret key. By out-of-band channel I mean a user typing a randomly chosen PIN number on both devices. Clearly, the security of a pairing phase is limited by a user's ability to choose good PIN numbers. In scenarios when one device in the pair does not have a keypad, security can be further compromised if the chosen PIN is transmitted to the other device in clear text.

Putting the Pieces Together

The ultimate objective of the Bluetooth specifications is to allow multivendor applications to interoperate. Different applications may run on different devices, and each device may use a protocol stack from one vendor and a Bluetooth chip from another. Yet interoperability among applications is achieved when different implementations comply with the same core and profiles specifications.

At the lowest layer, Bluetooth chips from different vendors interoperate over the air link because all Bluetooth chips implement the baseband and LMP specifications. Bluetooth stacks, which can be implemented as either firmware or software, include the L2CAP, SDP, and RFCOMM layers. It is relatively easy to port a Bluetooth stack from one platform to another because the lowest layer of a Bluetooth stack interfaces with a Bluetooth chip via a standard HCI interface which is also a part of the 1.1 specifications.

Porting a Bluetooth application from one stack to another, however, is more difficult. The application can use any standard API to access IP, PPP, OBEX, or RFCOMM layers of the Bluetooth stack, but there is no standard API to access the control functions provided by the Bluetooth stack. For example, if an application were to initiate a Bluetooth inquiry to discover other devices in its neighborhood, it must use an API specific to the stack vendor to access those functions.

Support for RFCOMM has been provided only for backward compatibility reasons. Legacy applications that run over serial cable, such as OBEX and PPP, will work over any Bluetooth stack without modifications. Thus, synchronization and IP-based applications already developed by vendors can be made available immediately when PDAs, cell phones, and laptops are Bluetooth enabled. The next release of Bluetooth specifications will provide better support for IP (without going through the PPP and RFCOMM layers), thus increasing portability of IP-based applications across all Bluetooth platforms. Standardization of control APIs, however, remains an unfinished task that has not yet been taken up by any standards organization.

Conclusion

Whether Bluetooth will live up to its promise or not will depend on a number of factors, some of which involve market forces rather than technical issues. For example, unless the initial adoption of Bluetooth is high, it will be difficult to meet the low-cost objective.

Security is also an open issue—as it is in almost all Internet applications. The free flow of information is desirable in some scenarios, but in general, proper safeguards are required to prevent the unauthorized leakage of information. The Bluetooth SIG is addressing the security issues associated with the initial usage scenarios, but new applications of Bluetooth will require a closer look at potential security threats.

One technical issue is that profile-based interoperability is easy to manage when the number of profiles is small, but market predictions indicate that more than a billion devices will be equipped with Bluetooth chips by 2005. This number is significantly greater than the number of hosts connected to the Internet today. As people find innovative uses of this technology, new profiles will be needed. Ensuring compliance with a rapidly increasing number of profiles will likely be difficult to maintain in the future. A good solution to this problem would be to quickly standardize an IP-over-Bluetooth specification since interoperability at the IP layer would automatically translate into interoperability at the applications layer. Some efforts have already begun within the Bluetooth SIG as well as in the IETF to resolve this issue.

Bluetooth has caught the attention of consumers because it would enable them do things that are otherwise cumbersome or not possible: synchronizing data between cell phones, laptops, and PDAs; using cell phones as cordless phones when at home; and connecting PDAs to the office LAN. The value proposition is therefore strong. The challenge is for vendors to meet these expectations.

References

1. *Specification of the Bluetooth System – Core*; available online at http://www.bluetooth.com/developer/specification/Bluetooth_11_Specifications_Book.pdf.
2. *Specification of the Bluetooth System – Profiles*; available online at http://www.bluetooth.com/developer/specification/Bluetooth_11_Profiles_Book.pdf.
3. G. Miklos et al., "Performance Aspects of Bluetooth Scatternet Formation," poster presentation at Mobile Ad Hoc Networks and Computing (MobiHOC 2000), IEEE/ACM workshop, Aug. 2000.
4. T. Salonidis et al., "Distributed Topology Construction of Bluetooth Personal Area Networks," *Proc. IEEE Infocom 2001*, IEEE Communication Society, New York, 2001.

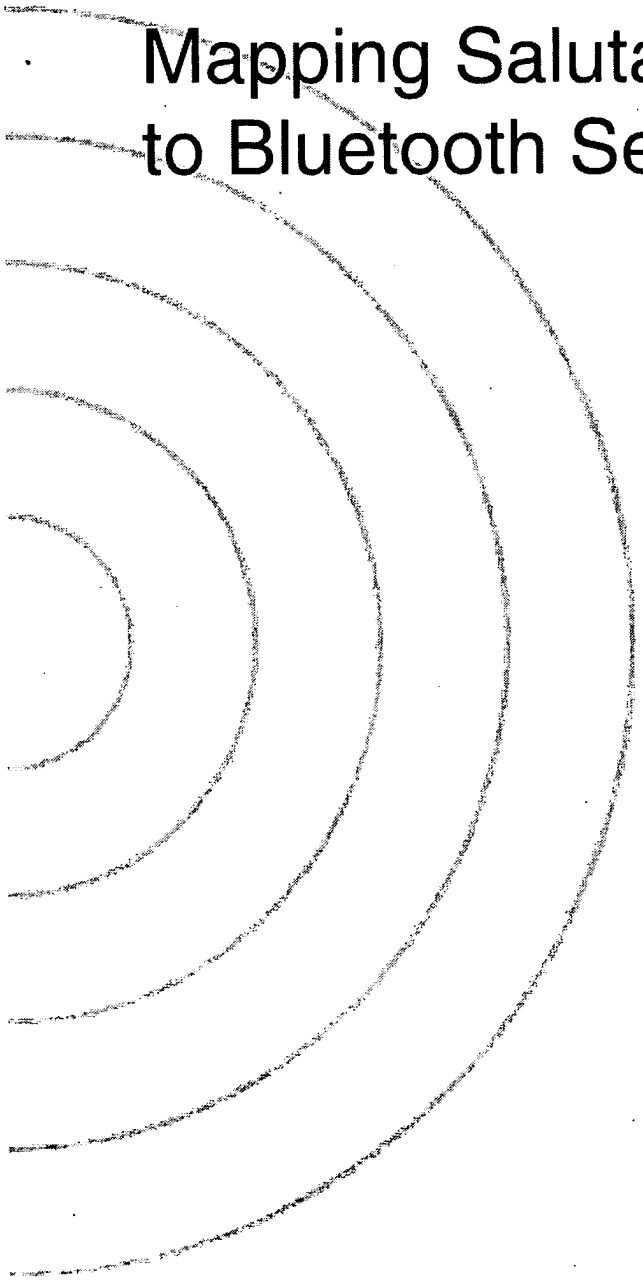
Pravin Bhagwat is the principal architect at Reefedge Inc., a networking infrastructure and software company that builds Bluetooth solutions for enterprise customers. He received a PhD in computer science from the University of Maryland, College Park. Bhagwat co-chaired the first Internet Engineering Task Force's BOF on IP over Bluetooth. He is chief architect of BlueSky, an indoor wireless networking system for palmtop computers, and co-inventor of TCP splicing, a technique for building fast application layer proxies.

Readers may contact the author at pravin@reefedge.com; http://www.cs.umd.edu/~pravin.

Bluetooth WHITE PAPER	DATE 01 July 99	N.B.	DOCUMENT NO 1.C.118/1.0
RESPONSIBLE Brent Miller	E-MAIL ADDRESS bamiller@us.ibm.com		STATUS

Mapping Salutation Architecture APIs to Bluetooth Service Discovery Layer

Version 1.0



Bluetooth provides a Service Discovery layer and defines a series of primitives to access the functions of the Service Discovery layer. This paper describes a mapping between these primitives and the APIs and functions in the Salutation Architecture.

Special Interest Group (SIG)

The following companies are represented in the Bluetooth Special Interest Group:
Ericsson Mobile Communications AB
IBM Corp.
Intel Corp.
Nokia Mobile Phones
Toshiba Corp.

Disclaimer and copyright notice

Revision History

Revision	Date	Comments
0.1	3/29/1999	First Draft
0.9	6/1/99	Response to SIG Comments
1.0	7/01/99	Final Version 1.0

Contributors

Robert Pascoe	Salutation Consortium
Brent Miller	IBM

Contents

1 Overview

The Bluetooth protocol stack contains a service discovery protocol (SDP) [1] that enables the retrieval of information that can be used to configure the stack to support several end-user applications. SDP can further be used to locate services that are available on devices in the vicinity of the user. Having located available services, a user may then select to use any of them.

SDP provides direct support for the following set of service inquires:

- search for services by service class;
- search for services by service attributes; and
- service browsing.

A service discovery profile is provided [2] that describes a generic syntax and semantics to be used by a service discovery application to locate services in other processes using Bluetooth SDP. The primitives are described in a generic way as these primitives may be operating environment dependent.

The Salutation Architecture [3] provides a standard method for applications, services and devices to describe and to advertise their capabilities to other applications, services and devices and to find out their capabilities. The architecture also enables applications, services and devices to search other applications, services or devices for a particular capability, and to request and establish interoperable sessions with them to utilise their capabilities.

This paper maps Bluetooth service discovery to the Salutation Architecture. Specifically this paper (1) maps the Bluetooth service discovery profile to the Salutation APIs and (2) maps the Bluetooth Service Discovery Protocol to the Salutation Manager.

2 Bluetooth Summary

2.1Bluetooth Service Discovery Protocol

Figure 2.1 shows the Bluetooth protocols and supporting entities.

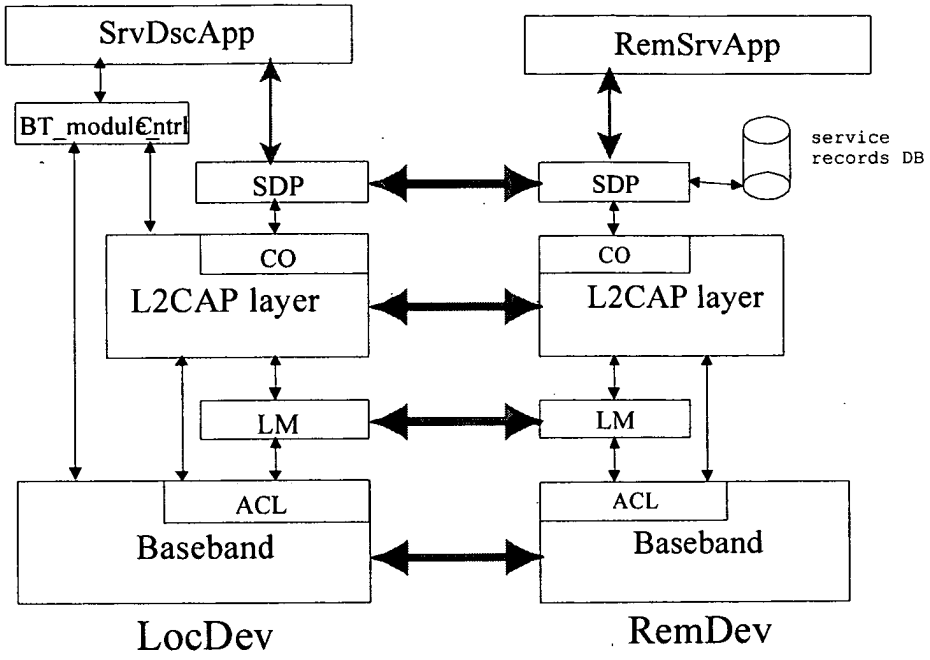


Figure 2.1: *The Bluetooth protocol for the service discovery profile*

The blocks marked SDP indicate the service discovery component. An SDP block generates and receives Bluetooth **service discovery protocol** (SDP) commands and responses from the lower layers of the Bluetooth stack.

SDP provides a means for client applications to discover the existence of services provided by server applications as well as the attributes of those services. The attributes of a service include the type or class of service offered and the mechanism or protocol information needed to utilise the service.

As shown in Figure 2.1, SDP involves communication between an SDP server located on RemDev and an SDP client located on LocDev. The server maintains a list of service records that describe the characteristics of services associated with the server. Each service record contains information about a single service.

A client may retrieve information from a service record maintained by the SDP server by issuing an SDP request.

All of the information about a service is maintained by an SDP within a single **service record**. The service record consists entirely of a list of **service attributes**. Each service attribute describes a single characteristic of a service.

SDP uses a request/response model where each transaction consists of one request **protocol data unit** (PDU) and one response PDU. Generally, each type of request PDU has a corresponding type of response PDU. However, if the server determines that a request is improperly formatted or for any reason the server cannot respond with the appropriate PDU type, it will respond with an SDP_ErrorResponse PDU.

2.1.1 Protocol Data Unit Types

- **ServiceSearch Transaction**:
  The SDP client generates a SDP_ServiceSearchRequest to locate service records that match the service search pattern given as the first parameter of the PDU. Upon receipt of this request, the SDP server will examine its service record data base and return an SDP_ServiceSearchResponse containing the service record handles of service records that match the given service search pattern.

- **ServiceAttribute Transaction:**
  The SDP client generates a SDP_ServiceAttributeRequest to retrieve specified attribute values from a specific service record. The service record handle of the desired service record and a list of desired attribute ids to be retrieved from that service record is supplied as parameters.

- **ServiceSearchAttribute Transaction:**
  The SDP_ServiceSearchAttributeRequest transaction combines the capabilities of the SDP_ServiceSearchRequest and the SDP_ServiceAttributeRequest into a single request. As parameters, it contains both a service search pattern and a list of attributes to be retrieved from service records that match the service search pattern. The SDP_ServiceSearchAttributeRequest and its response are more complex and may require more bytes than separate SDP_ServiceSearch and SDP_ServiceAttribute transactions. However, using SDP_ServiceSearchAttributeRequest may reduce the total number of SDP transactions, particularly when retrieving multiple service records.

- **Browsing for Services:**
  Normally, a client searches for services based on some desired characteristic(s) of the services. However, there are times when it is desirable to discover which types of services are described by an SDP server's service

records without any *a priori* information about the services. This process of looking for *any* offered services is termed browsing. In SDP, the mechanism for browsing for services is based on an attribute shared by all service classes. This attribute is called the *BrowseGroupList* attribute. Each attribute represents a *browse group* with which a service may be associated for the purpose of browsing. When a client desires to browse an SDP server's services, it creates a service search pattern containing the attribute that represents the *root browse group.*

Refer to Reference [1] for details about the Bluetooth service attribute definitions.

2.2 Bluetooth Profile Stack

Refering to Figure 2.1, the service discovery user application (SrvDscApp) in a local device (LocDev) interfaces with the SDP protocol to send service inquires and receive service inquire responses from other remote devices (RemDev). The SDP uses the connection-oriented (CO) transport service in L2CAP, which in turn uses the baseband asynchronous connectionless (ACL) links to carry ultimately the SDP PDUs over the air.

2.3 Bluetooth Service Primitives

This section briefly describes the service primitives that the Bluetooth stack needs to expose to the SrvDscApp to perform its task.

Table 2.1 contains a minimum set of enabling service primitives to support a SrvDscApp. Different implementations of the Bluetooth stack shall (at a minimum) enable the functions that these service primitives provide. For example, the **serviceSearch**() service primitive permits multiple identical operations to be handled at once. A stack implementation that requires an application to accomplish this function by iterating through the multiple identical operations one at a time will be considering as enabling the function of this service primitive.

service primitive	**function accomplished**
serviceBrowse (LIST(*RemDev*); LIST(*RemDevRelation*); LIST(*browseGroup*); *stopRule*)	searches for services (service browsing) that belong to the list of *browseGroup* services in the devices in the list of RemDevs; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched, e.g., search only the devices that are in the *RemDev* list for which pairing has been performed; search continues until the stopping rule *stopRule* is satisfied
serviceSearch (LIST(*RemDev*); LIST(*RemDevRelation*); LIST(*searchPath*,	searches whether the devices listed in the list of RemDevs support services in the requested list of services; each service in the list must have a service search path that is a superset of the *searchPath*; for each such service the values of the attributes contained

attributeList); *stopRule*)	in the corresponding *attributeList* are also retrieved; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched, e.g., search only the devices that are in the *RemDev* list for which pairing has been performed; search continues until the stopping rule *stopRule* is satisfied
enumerateRemDev (LIST(*classOfDevice*); *stopRule*)	searches for RemDev in the vicinity of a LocDev; RemDev searches may optionally be filtered using the list of *classOfDevice*, e.g., LAN APs; search continues until the stopping rule *stopRule* is satisfied
getRemDevName (LIST(*primitiveHandle*); *stopRule*)	retrieves the names of devices associated with the execution of the service primitives identified by the list of *primitiveHandle*;[1] search continues until the stopping rule *stopRule* is satisfied
terminatePrimitive (*primitiveHandle*; *returnResults*)	terminates the actions executed as a result of invoking the services primitive identified by the *primitiveHandle*; optionally, this service primitive may return any partially accumulated results related to the terminated service primitive

Table 2.1: Service primitives in support of SrvDscApp

[1] It is assumed that each invocation of a service primitive can be identified by a *primitiveHandle* the realization of which is implementation dependent.

3 Salutation Architecture Summary

The **Salutation Architecture** was created to solve the problems of ***service discovery and utilization*** among a broad set of appliances and equipment and in an environment of widespread connectivity and mobility.

The architecture provides a standard method for applications, services and devices to describe and to advertise their capabilities to other applications, services and devices and to find out their capabilities. The architecture also enables applications, services and devices to search other applications, services or devices for a particular capability, and to request and establish interoperable sessions with them to utilize their capabilities.

Given the diverse nature of target appliances and equipment in an environment of widespread connectivity, the architecture is processor, operating system, and communication protocol independent, and allows for scalable implementations, even in very low-price devices.

As shown in Figure 3-1, the Salutation Architecture defines an entity called the **Salutation Manager (SLM)** that functions as a service broker for applications, services and devices called a Networked Entity. The Salutation Manager allows Networked Entities to discover and utilize the capabilities of other Networked Entities.

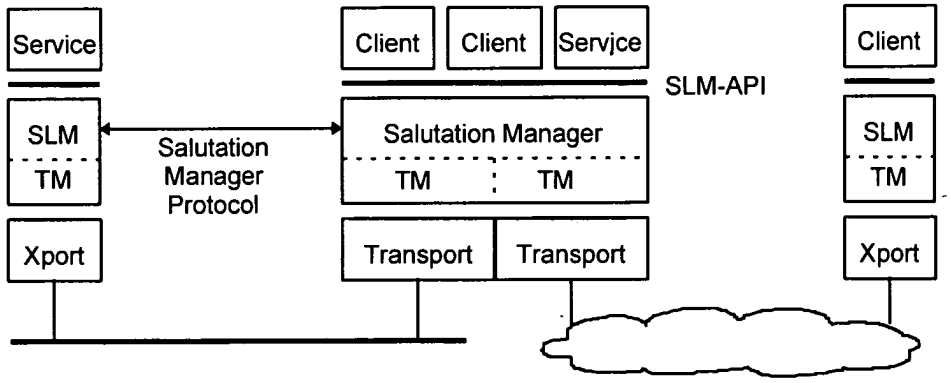


***Figure 3-1**: Model of the Salutation Manager*

A Networked Entity may be a service provider, called a **Service**. The Service registers its capability with a Salutation Manager. A Networked Entity may be a service user, called a **Client**. The Client discovers Services and requests to use them through a Salutation Manager. A Networked Entity may serve as either a Client or a Service, or both.

The Salutation Manager provides a transport-independent interface, called the **Salutation Manager Application Program Interface (SLM-API)**, to Services and Clients. The architecture defines an abstract procedural SLM-API.

The Salutation Manager communicates with other Salutation Managers to perform its role as a service broker. The Salutation Manager-to-Salutation Manager communications protocol is defined by the Salutation Architecture and called the **Salutation Manager Protocol**. The Salutation Manager binds to a specific transport through a **Transport Manager** (TM) unique to that transport class.

3.1 Service Broker Tasks

To perform its function as service broker, the Salutation Manager provides four basic tasks:

- Service Registry
- Service Discovery
- Service Availability
- Service Session Management

Service Registry and Service Discovery, and the APIs which expose these functions to the Client/Server layer, are of primary interest for this document.

3.1.1 Service Registry

The Salutation Manager contains a **Registry** to hold information about Services[2]. The minimum requirement for the Registry is to store information about Services connected to the Salutation Manager. Optionally, the Salutation Manager Registry may store information about Services that are registered in other Salutation Managers. All requests by other equipment for Salutation resources would be directed toward other Salutation Managers which would respond accordingly.

The limit on Registry implementation is the size of the storage reserved for the Registry function.

3.1.2 Service Discovery

The Salutation Manager can discover other remote Salutation Managers and determine the Services registered there. **Service Discovery** is performed by comparing a required Services type(s), as specified by the local Salutation Manager, with the Service type(s) available on a remote Salutation Manager. Through manipulation of the specification of required Service type(s), the Salutation Manager can determine:

- The characteristics of **all the Services** registered at a remote Salutation Manager

[2] Equivalent to Bluetooth service record DB shown in Figure 2.1

- The characteristics of a **specific Service** registered at a remote Salutation Manager
- The presence of a Service on a remote Salutation Manager **matching a specific set of characteristics**.

3.2 Salutation Manager API Specification

This section describes an abstract definition of the SLM-API, the application programming interface provided by the Salutation Manager to Salutation applications. More specifically, this section focuses on the Salutation Service Registration and Service Discovery APIs. It is intended to provide a level of understanding to aid in the mapping of Salutation APIs to Bluetooth SDP functions. The SLM-API supporting the mapping are:

- **Service Registration**

  slmRegisterCapability()

  slmUnregisterCapability()

- **Service Discovery**

  slmSearchCapability()

  slmQueryCapability()

3.2.1 Salutation Manager API Description

The SLM-APIs are described in the abstract in this section.

When the Client calls the Salutation Manager through the SLM-API, it is called the **local Salutation Manager**. Any other Salutation Manager is called a **remote Salutation Manager**. Refer to Reference [3] for details of the API attributes of SLM-ID, Functional Unit Description Record, and Service Description Record.

Abstract SLM-API	function accomplished
slmRegisterCapability() **Input Parameters** (Functional Unit Description Record[3]; Callback Entry for Open Service Indication; Callback Entry for Close Service	The **slmRegisterCapability()** function is called by Services to register their specific instances of Functional Units with the local Salutation Manager. The specific instance is described in a record called a Functional Unit Description Record. The calling Service passes a Functional Unit Description Record, which describes its capability, to the Salutation Manager. The Salutation Manager returns a Functional Unit Handle that uniquely identifies the Functional Unit among all the Functional Units registered with the Salutation Manager.

[3] The Salutation Architecture defines the Functional Unit Description Record as a record that identifies the Functional Unit, and the capabilities of that instance of the Functional Unit. The Functional Unit maps to the Bluetooth Service and capabilities map to the Bluetooth Attributes.

Indication; Callback Entry for Receive Data Indication; Preferred Functional Unit Handle) **Output Parameter** (Functional Unit Handle)	While a Service has a Functional Unit registered with a local Salutation Manager, the Functional Unit's capability may be included in the response to a *Query Capability* request. The Callback Entries are provided to provide an entry point into a Service when that service is to be used. Entry points are provided for opening and closing the Service as well as for receiving data. The Service may attempt to specify a handle for the functional unit instance being registered. This value will be assigned if it is not currently in use. Otherwise, the Salutation manager will assign a random, unused value for the handle.
slmUnregisterCapability() **Input Parameters** (Functional Unit Handle) **Output Parameter** None	The Service, which has registered itself with the local Salutation Manager by calling the **slmRegisterCapability()** function, calls this function to unregister itself from the local Salutation Manager. The Functional Unit Handle is the value returned by the ***slmRegisterCapability()*** used to register this Service.
slmSearchCapability() **Input Parameters** (SLM-ID; Service Description Record[4]; **Output Parameter** (List of SLM-IDs)	The Client calls this function to ask the local Salutation Manager to search for Salutation Managers having a registered Functional Unit with a specific capability. The local Salutation Manager returns the list of SLM-IDs to the Client. Salutation Manager(s) whose SLM-ID(s) are included in the list has(have) a Functional Unit(s) that can provide the Service requested by the Client. SLM-ID is NULL for version 2.0 of the Salutation Architecture Service Description Record describes the Service(s) and their capabilities that are of interest to the Client. A Service Description Record that contains a Functional Unit Description Record of "*All Call*" Functional Unit ID with no Attribute Records, may be specified to get the list of all the SLM-IDs of Salutation Managers known to the local Salutation Manager.
slmQueryCapability() **Input Parameters** (SLM-ID; Service Description Record)	The Client calls this function to discover registered Functional Units and their capabilities at a specific Salutation Manager. SLM-ID specifies the target Salutation Manager. If NULL is specified, the target Salutation Manager is the local Salutation Manager. The Input Service Description Record describes the Service(s) and their capabilities that are of interest to

[4] The Salutation Architecture defines the Service Description Record as a collection of one or more Functional Unit Description Records. The Service Description Record describes all the services sought by a Client or all the services maintained by a Service.

Output Parameter Service Description Record)	the Client. The Output Service Description Record describes the Service(s) and their capabilities that match the Input Service Description Record.

3.3 Service Discovery Flow

The flow of Remote Service Discovery messages and calls are depicted in Table 3-4. Salutation APIs are used by the Client and Functional Unit to access their respective Salutation Managers. Salutation Protocol flows between the Client-side and Server-side Salutation Managers.

Client	Client-side Salutation Manager	Salutation Protocol	Service-side Salutation Manager	Functional Unit
			<== slmRegisterCapability() call slmRegisterCapability() return ==>	
slmSearchCapability() call ==>				
	Query Capability call ==> <== Query Capability reply : (This step is repeated for each known SLM. The reply data maybe cached for the next step.)			
<== slmSearchCapability() return				
slmQueryCapability() call ==>				
	Query Capability call ==> <== Query Capability reply : (This step is optional, depending on the caching capability of the Client's Salutation Manager.)			
<== slmQueryCapability() return : (This step is repeated for each Salutation Manager found by the Search Capability. The Salutation Manager returns the cached data.)				
			<== slmUnRegisterCapability() call slmUnRegisterCapability() return ==>	

***Table 3-4**: Remote Service Discovery Flow Diagram*

4 Mapping Bluetooth SDP to Salutation APIs

Two approaches will be used for mapping Bluetooth SDP primitives to Salutation APIs.

The first approach will assume that the Salutation APIs are implemented on top of the Bluetooth service discovery. In this case, the mapping will show how SDP attributes can be passed in the Salutation APIs. That is:

Salutation APIs → Bluetooth SDP → Bluetooth Protocol

Here, the Salutation APIs are implemented as the entry to Bluetooth SDP. SDP extracts the information it requires from the APIs and processes according to the mapping to SDP primitives.

The second approach will assume that Salutation Manager can be map directly to the SDP protocol using a Bluetooth specific Transport Manager (indicated by TM in Figure 3.1). That is:

Salutation APIs → Salutation Manager → Bluetooth Protocol

Here, SDP is replaced by the Salutation Manager, with the Salutation Manager mapping its functionality to SDP protocol.

4.1 General Mapping Assumptions

- SDP is a service manager. For RemDev, it provides the ability to specify local services and respond to requests to discover the services it manages. For LocDev, it provides the ability for SvcDscApp to ask RemDev if it supports specific services. The APIs provide a means for application developers to access the functions of the SDP service manager.

- Service requests by LocDev are accessed through Salutation ***slmSearchCapability()*** and ***slm QueryCapability()*** API calls.

- Certain Bluetooth RemDevs will have the need to dynamically update the services they support. That is, a RemDev may need to update the service records maintained in the service record DB. Salutation ***slmRegisterCapability()*** and ***slmUnregisterCapability()*** API calls will

be mapped on the RemDev side to support dynamic registry of services.

4.2Salutation API Mapping

This section describes how Salutation APIs can be used to represent the SDP primitives.

4.2.1Configuration

The configuration used for this mapping is shown in Figure 4.1. The figure shows the use of the Salutation APIs as the entry point to SDP. No other changes have been made to the Bluetooth model shown in Figure 2.1.

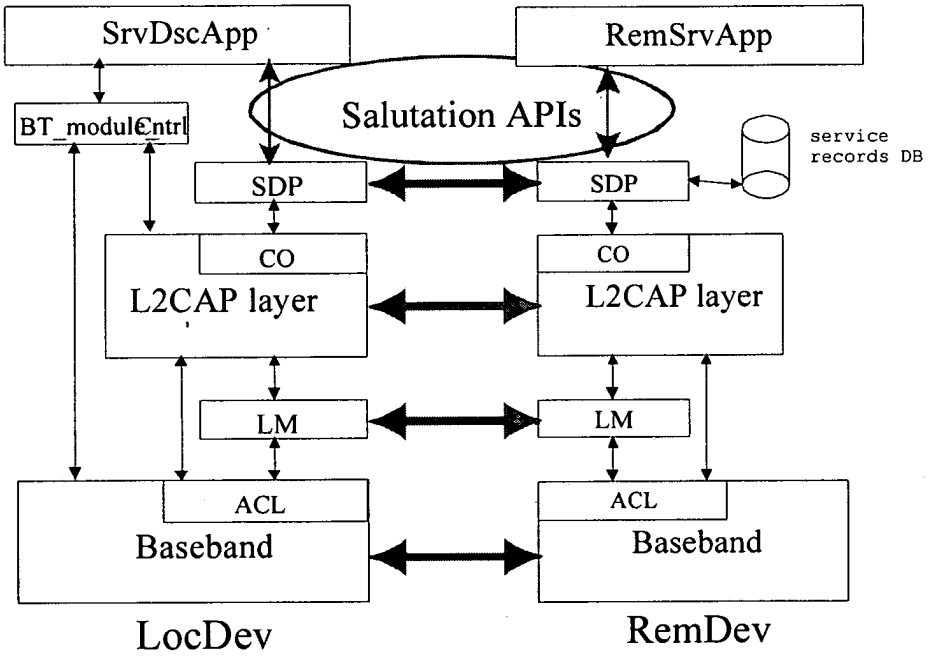


Figure 4.1: *Salutation API Mapping to Bluetooth SDP*

4.2.2Mapping

Table 4.1 depicts the general mapping of SDP primitives to Salutation APIs. Although there are no parallels defined in the SDP primitives, the Salutation Register Capability and Unregister Capability APIs are included for completeness.

The function provided by the SDP getRemDevName - the return of the names of devices identified by the SDP service searches - is an integral part of the Salutation's slmSearchCapability() and slmQueryCapability() calls. As a result, one Salutation API can provide the function of two SDP primitives.

This mapping uses the functional definition of the Salutation APIs, but not the parameter values. The APIs becomes a vehicle for passing the SDP parameters to the Bluetooth SDP Manager. That is, the SLM-ID and Service Description Records parameters of the Salutation APIs are replaced with the appropriate SDP parameter values. Therefore, the parameters defined by the SDP primitives are passed without modification to the Bluetooth SDP Manager via the Salutation API format. The parameter returned from the search operations is a list of the names of the devices identified by the search.

SDP Service Primitive	Salutation Primitive
ServiceBrowse, getRemDevName	**SlmQueryCapability()** **Input Parameters** (LIST(*RemDev*); LIST(*RemDevRelation*); LIST(*browseGroup*); *stopRule*) **Output Parameter** (List of Device Names)
ServiceSearch, getRemDevName	**SlmQueryCapability()** **Input Parameters** (LIST(*RemDev*); LIST(*RemDevRelation*); LIST(*searchPath*, *attributeList*); *stopRule*) **Output Parameter** (List of Device Names)
EnumerateRemDev, getRemDevName	**SlmSearchCapability()** **Input Parameters** (LIST(*classOfDevice*); *stopRule*

	) **Output Parameter** (List of Device Names)
(No SDP Registration Primitives)	**SlmRegisterCapability()** **Input Parameters** (LIST(*attributeList*); Callback Entry for Open Service Indication; Callback Entry for Close Service Indication; Callback Entry for Receive Data Indication; Preferred Functional Unit Handle) **Output Parameter** (Functional Unit Handle)
(No SDP Registration Primitives)	**slmUnregisterCapability()** **Input Parameters** (Functional Unit Handle) **Output Parameter** None

Table 4.1 *SDP primitive to Salutation API mapping*

The slmSearchCapability() call is used for both the serviceBrowse and the serviceSearch SDP primitives. The differentiater is the presence or absence of the browseGroup list parameter. If this parameter is present, the Bluetooth SDP Manager performs a browse operation. Otherwise a search operation is performed.

The Functional Unit Description Record parameter of the Salutation slmRegisterCapabilities API is replaced with the SDP attributeList parameter that specifies the capabilities of the service being registered. The callback parameters remain in the API definition, providing a means to define the entry points for service utilization. The returned value remains a handle of the registered service.

This value is used in the slmUnregiserCapability API to identify the service to be removed from public access.

4.2.3Summary

The Salutation API mapping provides a means to pass SDP primitive attributes to the Bluetooth SDP Manager.

4.3Salutation Manager Mapping

This section describes how the Salutation Manager, accessed via the Salutation APIs, can be used to generate Bluetooth SDP protocol.

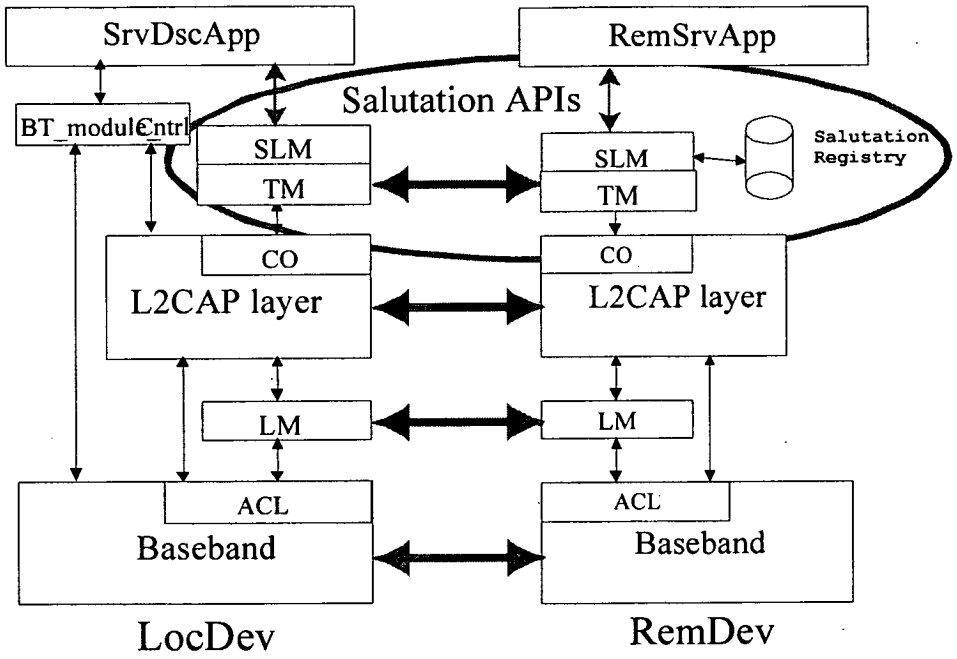


Figure 4.2: *Salutation Manager Mapping to Bluetooth SDP protocol*

4.3.1Configuration

The configuration used for this mapping is shown in Figure 4.2. The figure shows the use of the Salutation Manager (SLM) in both LocDev and RemDev to provide the service the management functionality of SDP. SLM exposes the existing Salutation APIs to the SrvDscApp and the RemSrvApp. SLM generates the appropriate SDP protocol through its TM, handing it off to L2CA layer. SLM also responds to SDP protocol received from L2CA. No other changes have been made to the Bluetooth model shown in Figure 2-1.

4.3.2Mapping

Table 4.2 depicts the general mapping of Salutation Manager functions to the SDP protocol. LocDev and RemDev use the Salutation APIs to access their respective Salutation Managers. Bluetooth Service Discovery Protocol flows between the LocDev and RemDev Salutation Managers.

Two transformations exist between the Salutation API layer and the Bluetooth SDP protocol layer. The Salutation Manager provides these transformations.

slmSearchCapability() ←→ SDP_ServiceSearch

slmQueryCapability() ←→ SDP_ServiceSearch & SDP_ServiceAttribute

4.3.2.1Capability Search

The Salutation slmSearchCapability() API call is mapped by the Salutation Manager's TM to a Bluetooth ServiceSearch protocol. The Salutation Manager maps the Functional Unit Description Record passed in the API to a *ServiceSearchPattern* attribute. The Salutation Manager then makes an L2CAP connection with a Bluetooth RemDev and sends the Service SearchPattern to the RemDev in a SDP_SereviceSearchRequest. Note that the ServiceSearchPattern may contain BrowseGroupList attributes as deemed necessary by the transform process.

If the RemDev Salutation Manager can match existing registered services with the ServiceSearchPattern, a SDP_ServiceSearchResponse is sent back to the LocDev Salutation Manager containing a list of service record handles for service records that match the ServiceSearchPattern in the request.

This process is repeated by establishing an L2CAP connection with another RemDev in radio range of LocDev. When all RemDevs have been contacted in this fashion, the LocDev Salutation Manager builds a list of device IDs returning positive responses and returns them to the calling application as a list of SLM-IDs. The Salutation Manager maintains a list of SLM-IDs and corresponding RemDev addresses for future SDP activity.

Specific transformations from Functional Unit Description Records to a ServiceSearchPattern will depend on the definition of ServiceSearchPattern for Bluetooth. For example:

LocDev	Client-side Salutation Manager	Bluetooth Service Discovery Protocol	Service-side Salutation Manager	RemDev
			<== slmRegisterCapability() call slmRegisterCapability() return ==>	
SlmSearchCapability() call ==>				
	SDP_ServiceSearchRequest ==> <== SDP_ServiceSearchResponse : (This step is repeated for each known Bluetooth device. The reply data maybe cached for the next step.)			
<== slmSearchCapability() return				
SlmQueryCapability() call ==>				
	SDP_ServiceSearchRequest ==> <== SDP_ServiceSearchResponse			
	SDP_ServiceSearchAttributeRequest ==> <== SDP_ServiceSearchAttributeResponse : (This step is repeated for each service record handle identified in the previous SDP_ServiceSearchRequest.)			
<== slmQueryCapability() return				
			<== slmUnRegisterCapability() call slmUnRegisterCapability() return ==>	

***Table 4-2**: Remote Service Discovery Flow Diagram*

if the following Service Classes are defined:

DuplexColorPostscriptPrinterServiceClass,
ColorPostscriptPrinterServiceClass,
PostscriptPrinterServiceClass,
PrinterServiceClass,

and the slmSerchCapability() call includes a [Print] Functional Unit Description Record with attributes of Postscript ,Duplex, and Colate,

then the resulting ServiceSearchPattern would contain attributes for PrinterServiceClass and PostscriptPrinterServiceClass.

The TM of the Salutation Manager on RevDev makes a reverse transformation from ServiceSearchPattern and compares to the registered Functional Unit attributes.

4.3.2.2 Capability Query

The Salutation slmQueryCapability() API call is mapped by the Salutation Manager to a Bluetooth ServiceSearch and ServiceAttribute protocols in a two step process.

1. The LocDev Salutation Manager's TM maps the Service Description Record passed in the API to a *ServiceSearchPattern* attribute. The Salutation Manager then makes an L2CAP connection with the Bluetooth RemDev represented by the SLM-ID passed in the API call, and sends the ServiceSearchPattern to the RemDev in a SDP_SereviceSearchRequest. Note that the ServiceSearchPattern may contain BrowseGroupList attributes as deemed necessary by the transform process.

   If the RemDev Salutation Manager can match existing registered services with the ServiceSearchPattern, a SDP_ServiceSearchResponse is sent back to the LocDev Salutation Manager containing a list of service record handles for service records that match the ServiceSeachrPattern in the request.

2. To determining attribute specifics, the LocDev Salutation Manager selects one of the service record handles returned in Step 1. The Salutation Manager's TM maps the Service Description Record passed in the API to an AttributeIDList attribute. The Salutation Manager then sends the service record handle and the AttributeIDList to the RemDev in a SDP_ServiceAttributeRequest.

   The RemDev returns a SDP_ServiceAttributeResponse containing an AttributeList identifying a list of attributes and their values for the requested service record.

   Step 2 is repeated for each service record handle returned in Step 1.

   When this cycle is completed, the LocDev Salutation Manager assembles a Service Description Record from the values returned by in the SDP_ServiceAttributeResponses. This Service Description Record is return to the calling application.

As before, specific transformations from the Functional Unit Description Records (contained in the Service Description Record) to a ServiceSearchPattern will depend on the definition of ServiceSearchPattern for Bluetooth. The same applies to transformations from Functional Unit Description Records to AttributIDList. For example:

if the following Service Classes are defined:

DuplexColorPostscriptPrinterServiceClass,
ColorPostscriptPrinterServiceClass,
PostscriptPrinterServiceClass,
PrinterServiceClass,

and the slmQueryCapability() call includes a [Print] Functional Unit Description Record with attributes of Postscript, Duplex, and Collate

then the resulting ServiceSearchPattern would contain attributes for PrinterServiceClass and PostscriptPrinterServiceClass, and the AttributeIDList would contain attribute IDs for Postscript, Duplex and Collate.

4.3.3 Stop Rules

Instances of the Salutation Manager, such as IBM's Salutation Manager Toolkit, provide a control interface outside of the Salutation APIs. This mapping assumes that such a control interface exists for the Salutation Manager supporting Bluetooth. The Bluetooth stop rules will be set via this interface.

4.3.4 Summary

The Salutation Manager mapping provides a means to use the Salutation Manager as a service broker in the Bluetooth environment. Because Salutation Manager is independent of underlying protocols and operating environments, a Salutation implementation can be a single application interface to numerous protocols. For example, in additions to the Bluetooth mapping, Salutation has been specified for TCP/IP and IR. A mapping to SLP is also being described.

As an example, Figure 4.3 shows a single service discovery application using the Salutation APIs to access the Salutation Manager to locate service in both the Bluetooth and TCP/IP environments. The advantage of this technique is to present a single methodology and API set to application for service discovery. The application need not know where a service resides, and therefore what service discovery primitives to uses, prior to performing a service search.

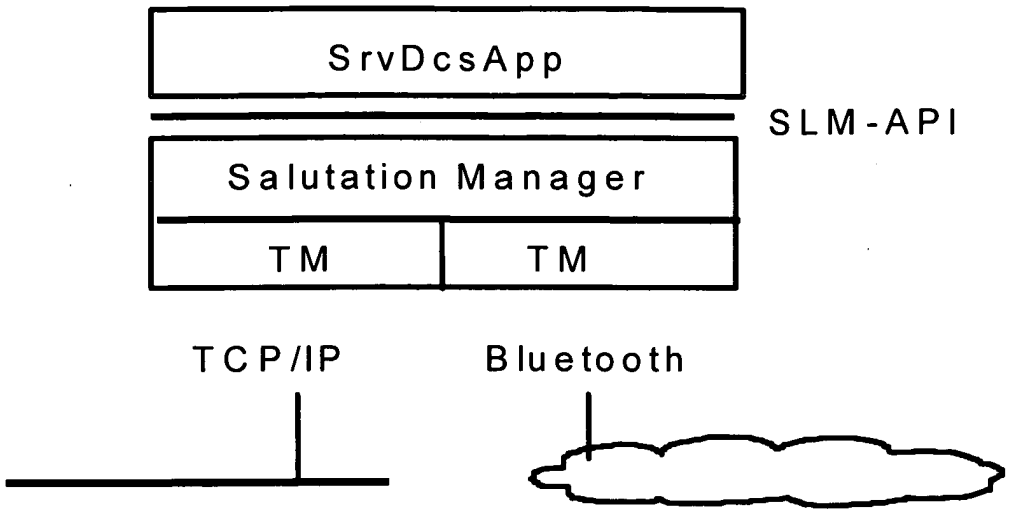


***Figure 4.3**: Salutation Manager in mixed transport environment*

5 References

[1] *Service Discovery Protocol*, Release 1.0 July 1999

[2] *Service Discovery Profile*, Release 1.0 July 1999

[3] *Salutation Architecture Specification (Part 1)* Version 2.0c (www.salutation.org)

6 Definitions

Term	Definition
Functional Unit Description Record	The Salutation Architecture defines the Functional Unit Description Record as a record that identifies the Functional Unit, and the capabilities of that instance of the Functional Unit. The Functional Unit maps to the Bluetooth Service and capabilities map to the Bluetooth Attributes.
Callback Entry for Open Service Indication	The specified entry of the calling Client is called back by the Salutation Manager when an *Open Service* request for the Functional Unit is received.
Callback Entry for Close Service Indication	The specified entry of the calling Client is called back by the Salutation Manager when a *Close Service* request for the Functional Unit is received.
Callback Entry for Receive Data Indication	The specified entry of the calling Client is called back by the Salutation Manager when a *Transfer Data* request for the Functional Unit is received.
Preferred Functional Unit Handle	If the calling Client wants to be assigned any specific value as its Functional Unit Handle, the preferred value is specified in this parameter. Otherwise, zero (0) should be specified.
Functional Unit Handle	The Salutation Manager generates a unique Functional Unit Handle value and returns it to the calling Client. If the function fails, zero (0) is returned.
SLM-ID	This parameter shall be NULL, indicating the local Salutation Manager, under version 2.0 of the Salutation Architecture.
Service Description Record	The Salutation Architecture defines a Service Description Record as a collection of one or more Functional Unit Description Records. The Service Description Record describes all the services sought by a Client or all the services maintained by a Service.

Security Comparison: Bluetooth™ Communications vs. 802.11

Thomas G. Xydis Ph.D.
Simon Blake-Wilson
Bluetooth Security Experts Group

11-10-2001
revised 5-15-2002

1.0 Introduction

Several attacks on IEEE 802.11b have been described in the media [1]. It has been shown that the WEP security framework used in IEEE 802.11 is susceptible to both attacks on data content and user authentication. These exposures allow an attacker to both inappropriately intercept data and also gain access to a network by impersonating a valid user.

Bluetooth and IEEE 802.11b are different, complementary technologies [10]. IEEE 802.11b is largely applied to LAN access, while Bluetooth LAN access is only one of many applications, most of which focus on smaller personal area networks (PANs). Different target applications and technology dictate different security architectures. With the differences between Bluetooth technology and IEEE 802.11b in mind, one may question the validity of comparing the security architectures of the two technologies. We feel, however, that such a comparison *is* valid. Indeed, from a user perspective the two technologies are really quite similar. Both are methods which allow computers to communicate to other devices, both use wireless technology, both operate in the 2.4 GHz spread spectrum band, etc. Due to these similarities, the public sometimes confuses Bluetooth communications with IEEE 802.11b. In addition, 802.11b security concerns have been unjustifiably applied to Bluetooth communications. However, these attacks do not apply to Bluetooth technology.

Here we discuss the two main attacks on 802.11b that have been described in the literature. We also explain why these attacks are not effective with Bluetooth wireless communications.

2.0 802.11b Eavesdropping

When a user sends data over a wireless network, he has a reasonable expectation that such data is not easily readable by unauthorized persons. Unlike a wired network, which requires a physical intrusion, wireless data packets can be received by anyone nearby with an appropriate receiver, potentially outside of the physical security barriers of an organization. This allows, so called

parking lot attacks, in which an attacker sits in a car in the parking lot of the intended victim. Accordingly, both Bluetooth and 802.11 technologies utilize data encryption in lower network layers.

The 802.11b specification utilizes a security framework called wireless equivalent privacy (WEP) protocol. A key component of WEP is the use of the stream cipher RC4. RC4 is a well-known and commonly used stream cipher, but its use in 802.11b is questionable owing to the nature of a wireless packet network.

RC4 operates by XORing the plaintext data with an encryption key stream. The result is commonly called ciphertext. RC4 is initialized with a secret WEP key and a public 24 bit long IV (initialization vector). If an attacker knows a plaintext and ciphertext pair *a-priori*, he can compute the encryption key stream using the XOR operation. Due to the low entropy of most plaintext messages, if an attacker can record a large number of ciphertext messages he can also compute the encryption key stream. For this reason users of RC4 are encouraged to change the encryption key on every message. The basic problem with RC4 over 802.11b is that wireless channels, will, by nature, occasionally lose data or drop packets. Thus, the synchronization between the encryptor and decryptor is difficult to maintain for any length of time. To overcome this limitation, WEP maintains synchronization by changing the 24-bit initialization vector (IV) on each packet. However 802.11b packets are relatively short. Therefore, one can expect a key/IV combination to be repeated every few seconds. Therefore one can expect a key/IV combination to be repeated relatively frequently. For more details, see Walker [2], and Borisov et al [3].

Furthermore, there are more sophisticated attacks that exploit subtle properties of the key-scheduling algorithm of RC4 deployed with WEP. In July 2001, Fluhrer, Mantin, and Shamir [4] showed how to exploit the particular way that 802.11b derives the encryption key stream from the initialization vector IV and the shared key to completely recover the secret key. The paper shows how to recover the key, byte by byte, by exploiting a bias in the statistical distribution of the candidate secret keys that becomes manifest when observing the first byte of the key stream. By feeding the system with sufficiently many publicly known IV strings, the authors show how this bias can be exploited to correctly determine the key, one byte at a time. A team of AT&T researchers [2] soon after demonstrated a practical implementation of the attack using off-the-shelf equipment. The implementation showed that the key could be recovered by eavesdroppinging roughly 5 million encrypted packets. Thus WEP encryption is relatively easy to break.

3.0 802.11b False Authentication

To gain access to a network, a user must be authenticated. While authentication is typically done at a higher network level, 802.11b and Bluetooth technologies also support device authentication.

In 802.11b authentication is performed by a challenge response procedure using a shared secret. After requesting authentication, the authenticator sends the initiator a 128-octet random number challenge. The initiator encrypts the challenge using the shared secret and transmits it back to the authenticator. Encryption is performed by XORing the challenge with a pseudo-random string formed by the shared secret and a public IV. Note that the only thing that changes from authentication to authentication with a specific user is the plaintext message. [1]

A simple and powerful attack on this authentication mechanism is presented by Arbaugh, et. al [5]. First the intruder determines the pseudorandom string by recording the challenge (plaintext) and the response (ciphertext) and XORing them. He then impersonates the victim by using the pseudorandom string to compute the response to subsequent challenges. Notice that the attacker never needs to determine the shared secret; knowledge of the pseudorandom string is sufficient.

4.0 Device Authentication in Bluetooth Technology

Like 802.11b, Bluetooth technology provides a method for authenticating devices. Device authentication is provided using a shared secret between the two devices. The common shared secret is called a *link key.* This link key is established in a special communications session called pairing. All paired devices (devices that have had a previous connection to establish security procedures) share a common link key. There are two types of link keys defined in the [9]: *unit keys* and *combination keys.*

A device using a unit key uses the same secret for *all* of its connections. Unit keys are appropriate for devices with limited memory or a limited user interface. During the pairing procedure the unit key is transferred (encrypted) to the other unit. Note that only one of the two paired units is allowed to use a unit key.

Combination keys are link keys that are unique to a particular *pair* of devices. The combination key is only used to protect the communication between these two devices.

Clearly a device that uses a unit key is not as secure as a device that uses a combination key. Since the unit key is common to all devices with which the device has been paired, all such devices have knowledge of the unit key.

---

[1] 802.11 also supports "Open System Authentication" which is essentially no authentication, in other words, everyone who requests an authentication is authenticated. In our view, this is not a security flaw. The implementation choice to not authenticate is a valid one in some situations, such as where security requirements are limited or authentication is provided at a higher network level. A similar choice is allowed in the Bluetooth wireless communications.

Consequently they are able to eavesdrop on any traffic based on this key. In addition, they could, in theory, be modified to impersonate other devices using the key. Thus, when using a unit key there is no protection against attacks from other devices with which the device has been paired. As a result, the Bluetooth SIG discourages the use of unit keys in secure applications.

Authentication is performed with a challenge response scheme utilizing the E1 algorithm. E1 is a modification of the block cipher SAFER+. The scheme operates as follows: The verifier issues a 128 bit long challenge. The claimant then applies E1 using the challenge, its 48-bit Bluetooth address, and the current link key. He then returns the 32 most significant bits of the128 bit result[2]. The verifier confirms the response, in which case the authentication has succeeded. In this case, the roles are switched and the same procedure is applied again, thereby accomplishing mutual authentication.

The Bluetooth challenge response algorithm differs from that used in 802.11b in very important ways. In 802.11b the challenge and response form a plaintext/ciphertext pair. This fact, combined with the simplicity of the encryption method (XOR), allow an intruder to easily determine the authentication key string by listening to one authentication procedure. In contrast, the Bluetooth authentication method never transmits the complete challenge response pair. In addition, the E1 algorithm is not easily invertible. Thus even if an attacker has recorded an authentication challenge response session, he cannot (directly) use this data to compute the authentication key.

5.0 Data Eavesdropping, 802.11

The Bluetooth standard does not use RC4 but rather the stream cipher E0, which is specifically designed to run over a Bluetooth wireless packet network. A unique encryption key is generated for each session, from which per-packet keys are derived, in a manner that avoids the problem in 802.11b caused by frequent reuse of per-packet keys.

Direct attacks on the E0 cipher are known but are of significant complexity. Jakobsson and Wetzel present two such attacks [6] the first is of 2^{100} complexity, the second is a "birthday-type attack" of 2^{66}overall complexity. Fluher and Lukas [7] present an attack using observed keystream and the public knowledge of the encryption mechanism used in E0 to compute the encryption key. Their method requires from $O(2^{73})$ to $O(2^{84})$, depending on how much cleartext is available for the algorithm. They contend that the upper limit of E0 is actually about 80 bits and question the extension of the E0 key size to 128 bits as suggested in the Bluetooth specification [8]. As discussed by Jakobsson and Wetzel [6], attacks with a high order of complexity are not of practical value, but may point the way

[2] The remaining bits are called the Authentication Ciphering Offset (ACO) and are used to derive the ciphering key for data encryption.

to a more efficient attack. As yet, a more efficient direct attack on E0 has not been reported.

Like RC4, E0 required a ciphering key. The ciphering key is computed as a hash of a random number, the link key and a byproduct of the authentication procedure the Authentication Ciphering Offset (ACO).

While the link key is also used to generate a ciphering key used for data encryption, it is not used for data encryption itself. This is a significant advantage over 802.11b in which the same key is used for authentication and encryption.

In summary the known attacks on the E0 cipher used in Bluetooth are far more computationally complex then corresponding attacks on RC4 used in 802.11b. As yet, no practical direct attack has been reported. Also, unlike 802.11b, different keys are used for authentication and encryption. Accordingly practical studies on Bluetooth security have been focused on methods to guess or steal the key (or at least a portion of it). The most logical time to attempt this is during the pairing procedure.

6.0 Bluetooth Pairing

As discussed in Section 4.0 pairing is the procedure where a relationship (link key) is established between two previously unknown devices. The link key is *derived* when the devices are initially paired (i.e. the link key does not exist before the pairing procedure). Pairing is facilitated with yet another key, the *initialization* key. This key is computed by a pair of devices using the Bluetooth addresses of each device, a random number, and a shared secret (PIN). Since it is only used in the initial pairing, the initialization key is only used once.

The initial pairing is the most profitable area of attack on a Bluetooth device. If the attacker can guess or steal the PIN during the initial pairing, then he can perform a much more efficient search to derive the link key. This search is further simplified if the communications occurring while the devices are paired is recorded [6]. For this reason the Bluetooth SIG strongly encourages the use of long, random PINs and suggests that pairing be performed only in a private place. Assuming that both devices have a man-machine interface (such as a keypad) it is also suggested that the PIN be manually entered into both devices or in any case communicated out-of-band (not transmitted over the Bluetooth wireless link). Thus, long PINs provide improved security since the PIN cannot be received over-the-air. To steal the PIN an attacker must guess or record it by some other means such as direct observation of the user, a more difficult procedure if the PIN is long and the pairing is performed in private.

7.0 Final Comments

The known attacks on 802.11b security have been discussed and found not to apply to Bluetooth wireless technology. In particular

a) 802.11b authentication is highly susceptible to impersonation by recording only one authentication procedure. This is facilitated because a plaintext/ciphertext pair is transmitted. Bluetooth communications do not share this limitation.
b) 802.11b encryption is not very secure. The RC4 implementation used in 802.11b has several well-known direct attacks. Currently known direct attacks on the Bluetooth encryption are computationally complex and of little practical value.

From the preceding discussion it is clear that the weakest link in the Bluetooth security architecture is the initial pairing especially if a weak PIN is used. Accordingly the Bluetooth SIG strongly encourages pairing in a private place and the use of robust PINs. In addition, simple devices that use unit keys should not be relied upon to communicate highly secure data.

As a communication standard, Bluetooth security focuses on the link level. It provides both entity authentication and link privacy. Since these functions are focused at the lower network layers, message authentication and secure end-to-end links are not provided. However, many applications, such as e-mail and browser transactions require end-to-end security. As with other communication standards, this function is expected to be provided at higher network layers by specific application providers. Accordingly, the Bluetooth SIG encourages the reuse of existing transport, session and application layer security.

Regarding the security limitations that have been reported for 802.11b; the WLAN community is currently examining these issues. We expect them to be resolved with subsequent revisions of the standard. In addition several 802.11b vendors have added proprietary authentication and encryption procedures at higher network layers.

Acknowledgements

The authors wish to thank the members of the Bluetooth Security Experts Group and also Brent Miller and Peter Lee from IBM for their helpful comments.

References

[1] W. A. Arbaugh, "Wireless Research", available from, http://www.cs.umd.edu/~waa/wireless.html

[2] J. Walker, "Unsafe at any key size; An analysis of the WEP encapsulation", available from http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/0-362.zip

[3] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11." available from http://www.issac.sc.berkley.edu/issac/wep-faq.html.

[4] S. Fluhrer, I. Mantin, A. Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, in the Workshop Record of SAC 2001

[5] W. Arbaugh, N. Shankar, Y.C.J. Wan, "Your 802.11 Wireless Network has No Clothes" available from http://www.cs.umd.edu/~waa/wireless.pdf

[6] M. Jakobsson and S. Wetzel, "Security Weaknesses in Bluetooth" available from http://www.rsasecurity.com/rsalabs/staff/bios/mjakobsson/bluetooth/bluetooth.pdf

[7] S. Fluhrer and S. Lucks, "Analysis of the E0 Encryption System" available from S. Lucks' web site at http://th.informatik.uni-mannheim.de/People/Lucks/papers/e0.ps.gz , a gnu-zipped Postscript file.

[8] Bluetooth SIG, *Specification of the Bluetooth system, Profiles",* Version 1.1, 1 February 22, 2001, available at http://www.bluetooth.com/.

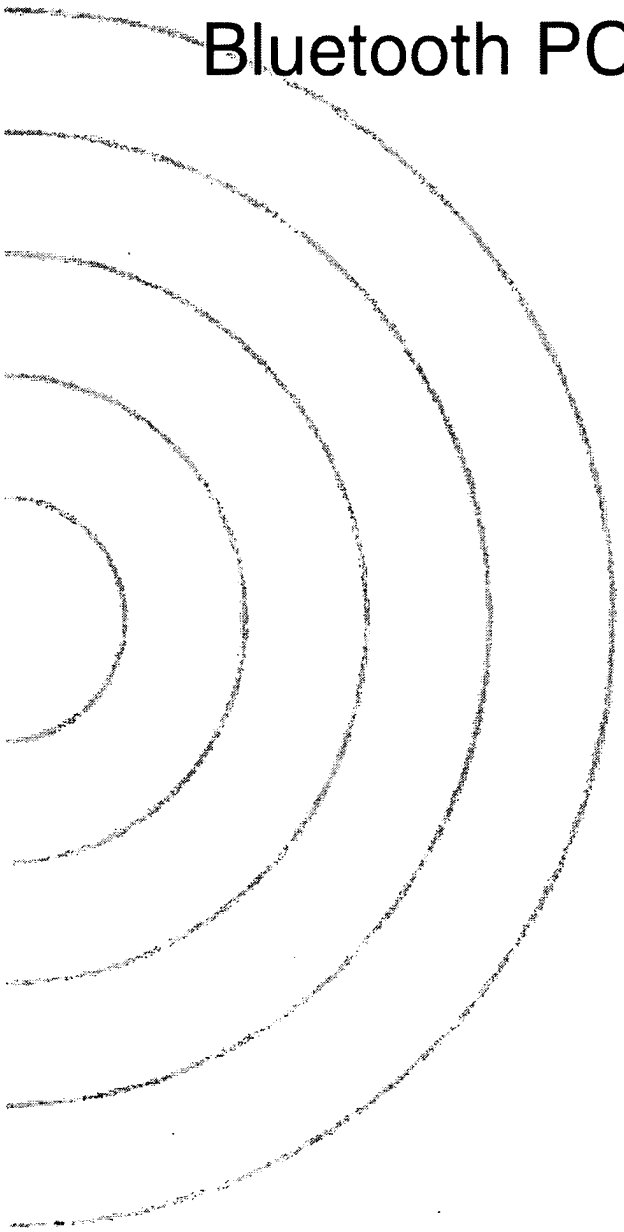
[9] Bluetooth SIG, *Specification of the Bluetooth system, Core ",* Version 1.1, 1 February 22, 2001, available at http://www.bluetooth.com/.

[10] B. Miller, "IEEE 802.11 and Bluetooth wireless technology" available from http://www-106.ibm.com/developerworks/wireless/library/wi-phone/

Bluetooth WHITE PAPER	DATE 25 August 99	N.B.	DOCUMENT NO 1.C.123/1.0
RESPONSIBLE Riku Mettala	E-MAIL ADDRESS Riku.Mettala@nmp.nokia.com		STATUS

Bluetooth PC Card Transport Layer

Version 1.0



The Bluetooth module can be connected to the Bluetooth host, e.g., a PC in various ways. One possibility is to place the module into a PC Card and the required communications is done between the PC and the PC Card.

Special Interest Group (SIG)

The following companies are represented in the Bluetooth Special Interest Group:
Ericsson Mobile Communications AB
IBM Corp.
Intel Corp.
Nokia Mobile Phones
Toshiba Corp.

Contributors

Cooklev, Todor	3Com Corporation
Inouye, Jon	Intel Corporation
Mettälä, Riku	Nokia Mobile Phones

Disclaimer and copyright notice

Contents

1 Overview

Bluetooth SIG (Special Interest Group) has defined that USB, RS232 (serial cable), UART, and PC Card are alternatives for a Bluetooth module to be connected with a PC (host). Bluetooth SIG has specified the standardized interfaces for the USB, RS232, and UART but the interface to the PC Card will not be standardized. The reasons for this are varied but the dependency of the interface on the PC Card implementation and the fact, that SIG does not want restrict the technology by standardizing the interface, are examples.

The purpose of this white paper is to describe the general functionality how the PC Card communicates with the host and the general requirements for the SW component delivered within the Bluetooth PC Card product. However, the programming interface between the module within the PC Card and the host will not be specified according to the Bluetooth SIG decision. Manufacturers should use existing PC Card and Cardbus standards for designing their products along with existing PC and PDA guidelines regarding cards designed for these types of devices.

Figure 1 depicts the components of the Bluetooth host and the Bluetooth module in the cases of the USB, RS232, and PC Card devices. UART is not included in the figure. The upper interfaces of the minidriver (USB, RS232, or PC Card minidriver) are the interfaces, which connect the transport layer to the rest of the Bluetooth SW protocol stack on the host. This interface (See also Chapter 3.1) and the lower interface of the HC driver comply with each other and so, a proper information can exchange between these drivers. However, the HC driver may not have any knowledge, which of the transport layer connects the host and the module.

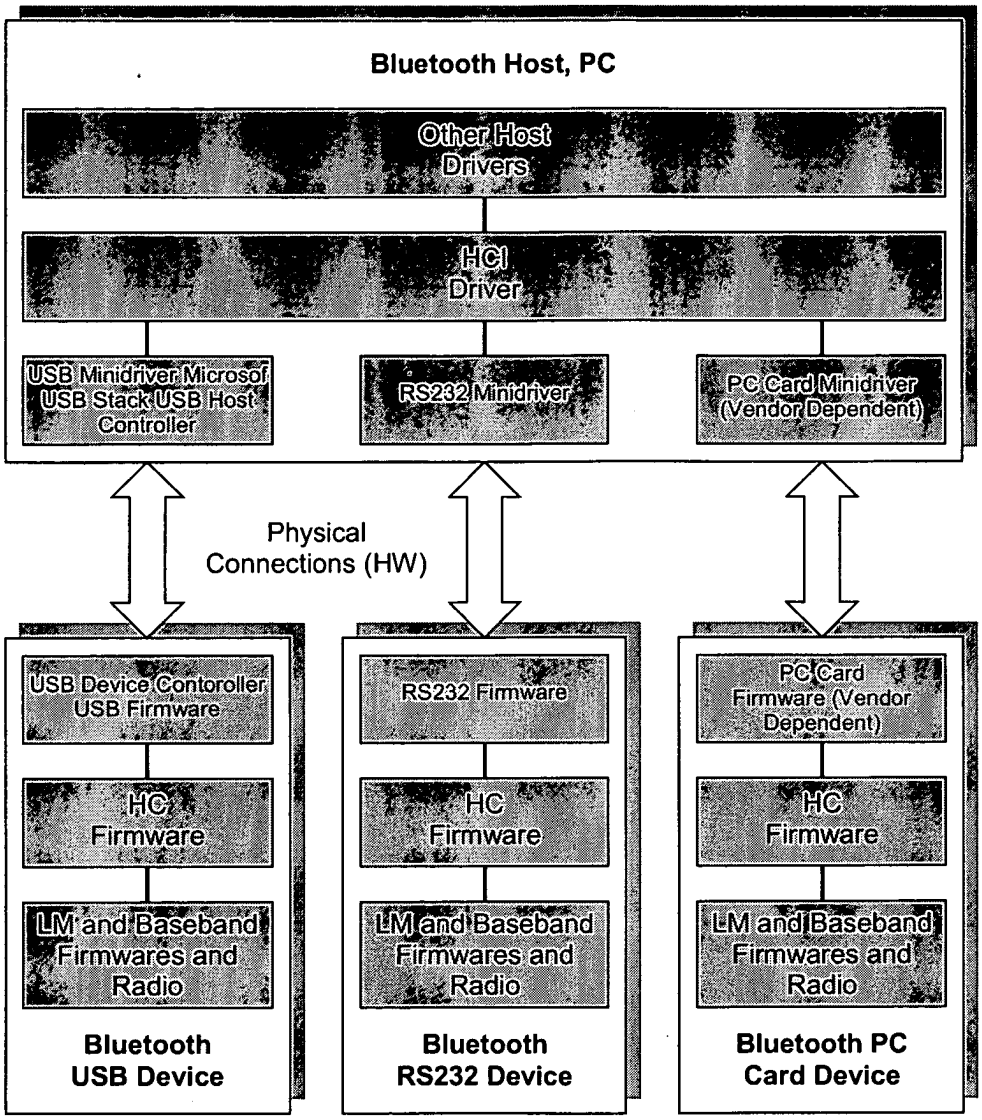


Figure 1 USB, RS232, and PC Card Transport Layers

2 Transport Functionality

The PC Card transport layer indicates and transfers the different types of HC packets using the physical bus from the host to the module and vice versa. Thus, the receiving end is able to separate the different packet types. Figure 2 depicts this procedure.

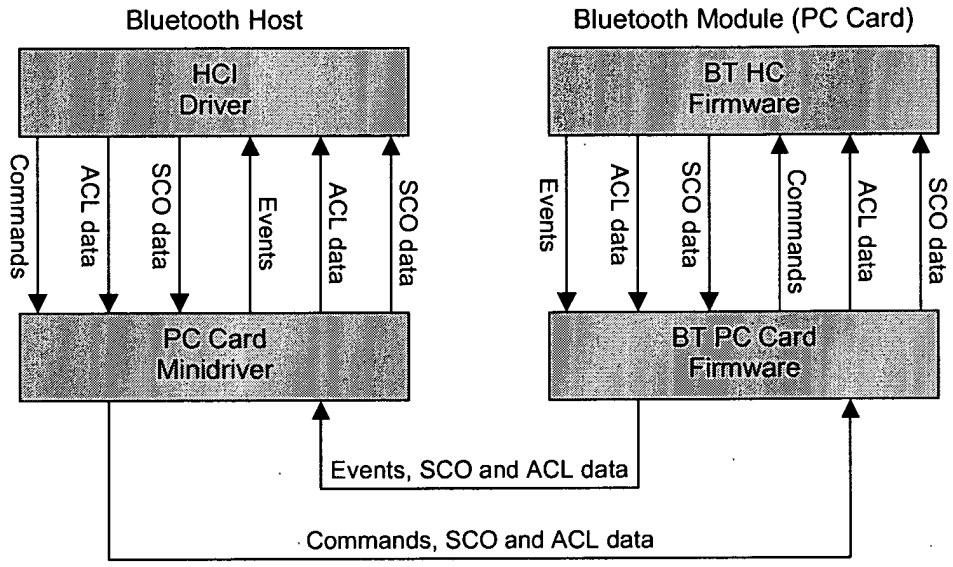


Figure 2 Bluetooth PC Card Transport Layer Functionality

2.1 Packet Types over Transport Layer

There are at least four different packet types, which the transport protocol must recognize (See also Figure 2). These packet types are HCI commands, HCI events, ACL data, and SCO data. The transport layer (PC Card Driver or Bluetooth PC Card Firmware) has to get information about the packet types directly from the HCI driver or from the HC firmware. Otherwise, the transport layer is not able to forward the packet type information from the host to the module or vice versa. However, the PC Card transport layer has not to have any visibility into the payload data, which the HCI driver sends to the host controller in the module or vice versa.

This document does not specify how the transport layer indicates the separate packet types. This depends on the different PC Card solutions.

3 PC Card Minidriver Requirements

The purpose of the Bluetooth PC Card minidriver is to give a possibility for sending and receiving the HCI messages to and from the PC Card, respectively. The minidriver has to provide two interfaces such that this communication is possible. These interfaces are the interface with the HCI driver (the upper interface) and the interface with the physical bus (the lower interface), to which the PC Card is also physically connected.

3.1 Interface with HCI Driver

If the vendor implementing a Bluetooth PC Card and a Bluetooth PC Card minidriver also delivers the rest of the Bluetooth SW protocol stack, there are no external requirements to this interface and the interface can be vendor specific. Also, in this case a separate Bluetooth PC Card minidriver may not exist but it is integrated with the other drivers including the HCI driver.

If the vendor delivers only the Bluetooth PC Card and the minidriver for it, the interface must comply with the lower interface of the HCI driver of an existing Bluetooth SW protocol stack implementation.

3.2 Interface with Physical Bus

There are no requirements for the lower interface of the Bluetooth PC Card minidriver. It is assumed that the Bluetooth PC Card including the PC Card firmware and the minidriver are provided by the same vendor. Thus, there will be no interoperability problems with the communications between the firmware and minidriver.

4 References

[1] Bluetooth Special Interest Group, Bluetooth Host Controller Interface Functional Specification

[2] Bluetooth Special Interest Group, Bluetooth HCI USB Transport Layer Specification

[3] Bluetooth Special Interest Group, Bluetooth HCI RS232 Transport Layer Specification

[4] Bluetooth Special Interest Group, Bluetooth UART Transport Layer Specification

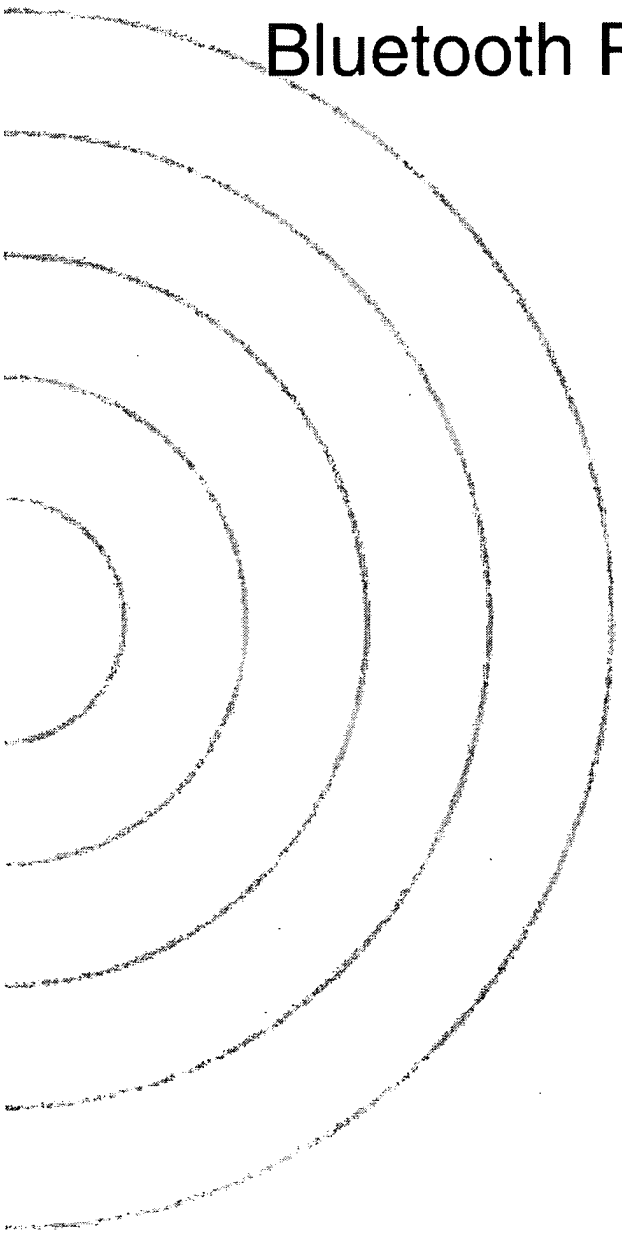
5 Acronyms

Abbreviation or Acronym	Meaning
ACL	Asynchronous ConnectionLess
HC	Host Controller
HCI	Host Controller Interface
LM	Link Manager
PDA	Personal Digital Assistant
SCO	Synchronous Connection-Oriented
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

Bluetooth WHITE PAPER	DATE Aug 25th 99	N.B.	DOCUMENT NO 1.C.120/1.0
RESPONSIBLE Riku Mettala	E-MAIL ADDRESS riku.mettala@nmp.nokia.com		STATUS

Bluetooth Protocol Architecture

Version 1.0



This white paper describes the protocol architecture developed by the Bluetooth Special Interest Group (SIG). Various usage models are presented and complemented with a description of the protocols relevant to their implementation.

Special Interest Group (SIG)

The following companies are represented in the Bluetooth Special Interest Group:
Ericsson Mobile Communications AB
IBM Corp.
Intel Corp.
Nokia Mobile Phones
Toshiba Corp.

Contributors

Bisdikian, Chatschik	IBM Corporation
Bouet, Stephane	Nokia Mobile Phones
Inouye, Jon	Intel Corporation
Mettälä, Riku	Nokia Mobile Phones
Miller, Brent	IBM Corporation
Morley, Ken	3Com Corporation
Muller, Thomas	Nokia Mobile Phones
Roter, Martin	Nokia Mobile Phones
Slotboom, Erik	Ericsson Mobile Communications AB

Disclaimer and copyright notice

Contents

1 Introduction

The Bluetooth Special Interest Group (SIG) has developed the Bluetooth Specification Version 1.0 Draft Foundation (thereafter to be referred to as the "Specification"), that allows for developing interactive services and applications over interoperable radio modules and data communication protocols. The objective of this paper is to provide an overview of the protocols in the Specification, their capabilities and the relation to each other (referred to as the "Bluetooth protocol architecture"). Moreover, a number of usage models identified by the Bluetooth SIG will be presented and it will be shown how (and which of) these protocols are stacked to support these usage models.

1.1 Bluetooth Protocol Stack

The ultimate objective of the Specification is to allow applications written in a manner that is conformant to the Specification to interoperate with each other. To achieve this interoperability, matching applications (e.g., corresponding client and server application) in remote devices must run over identical protocol stacks. The following protocol list is an example of a (top-to-bottom) protocol stack supporting a business card exchange application: vCard → OBEX → RFCOMM → L2CAP → Baseband. This protocol stack contains both an internal object representation convention, vCard, and "over-the-air" transport protocols, the rest of the stack.

Different applications may run over different protocol stacks. Nevertheless, each one of these different protocol stacks use a common Bluetooth data link and physical layer, see more details on the protocol layers in the next section. Figure 1 shows the complete Bluetooth protocol stack as identified in the Specification on top of which interoperable applications supporting the Bluetooth usage models are built. Not all applications make use of all the protocols shown in Figure 1. Instead, applications run over one or more vertical slices from this protocol stack. Typically, additional vertical slices are for services supportive of the main application, like TCS Binary (Telephony Control Specification), or SDP (Service Discovery Protocol). It is worth of mentioning that Figure 1 shows the relations how the protocols are using the services of other protocols when payload data needs to be transferred over air. However, the protocols may also have some other relations between the other protocols. E.g., some protocols (L2CAP, TCS Binary) may use LMP (Link Manager Protocol) when there is need to control the link manager.

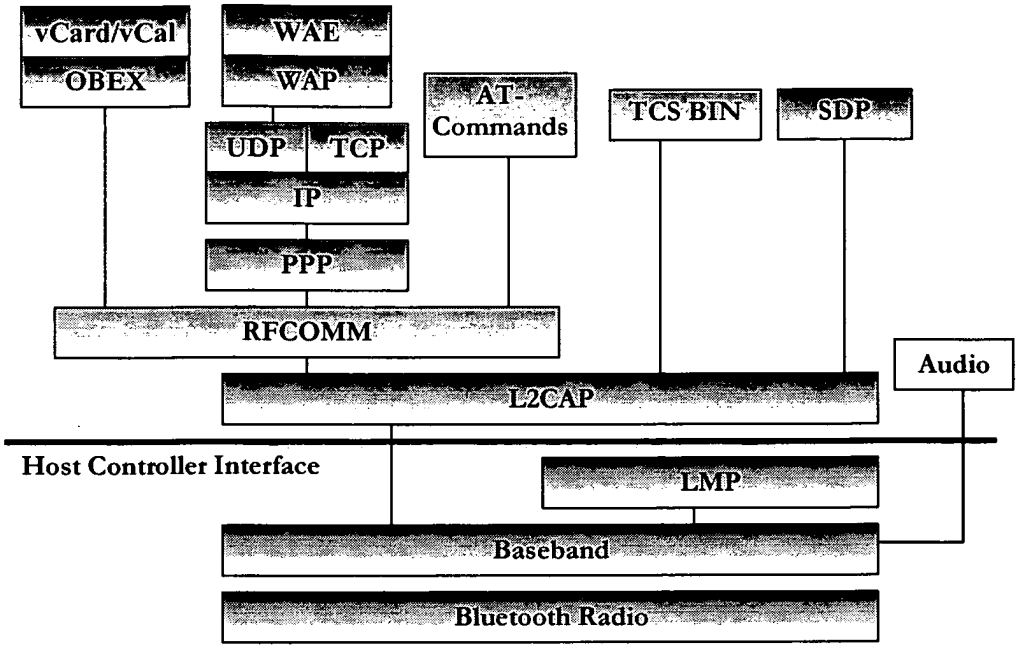


Figure 1 Bluetooth Protocol Stack

As seen in Figure 1, the complete protocol stack comprises of both Bluetooth-specific protocols like LMP and L2CAP, and non-Bluetooth-specific protocols like OBEX (Object Exchange Protocol) and UDP (User Datagram Protocol). In designing the protocols and the whole protocol stack, the main principle has been to maximize the re-use of existing protocols for different purposes at the higher layers, instead of re-inventing the wheel once again. The protocol re-use also helps to adapt existing (legacy) applications to work with the Bluetooth technology and to ensure the smooth operation and interoperability of these applications. Thus, many applications already developed by vendors can take immediate advantage of hardware and software systems, which are compliant to the Specification. The Specification is also open, which makes it possible for vendors to freely implement their own (proprietary) or commonly used application protocols on the top of the Bluetooth-specific protocols. Thus, the open Specification permits the development of a large number of new applications that take full advantage of the capabilities of the Bluetooth technology.

2 Protocols in Bluetooth Architecture

The Bluetooth protocol stack can be divided into four layers according to their purpose including the aspect whether Bluetooth SIG has been involved in specifying these protocols. The protocols belong into the layers in the following way.

Protocol layer	Protocols in the stack
Bluetooth Core Protocols	Baseband [1], LMP [2], L2CAP [3], SDP [4]
Cable Replacement Protocol	RFCOMM [5]
Telephony Control Protocols	TCS Binary [6], AT-commands [7],[8],[9]
Adopted Protocols	PPP [10], UDP/TCP/IP [10], OBEX [11], WAP [12], vCard [13] , vCal [14], IrMC[1] [15], WAE [16]

Table 1: The protocols and layers in the Bluetooth protocol stack

In addition to the above protocol layers, the Specification also defines a Host Controller Interface (HCI), which provides a command interface to the baseband controller, link manager, and access to hardware status and control registers. This interface is not discussed further in this paper, but more information can be obtained from the functional specification of Bluetooth host controller interface [17]. In Figure 1, HCI is positioned below L2CAP but this positioning is not mandatory but HCI can exist e.g., above L2CAP.

The Bluetooth Core protocols comprise exclusively Bluetooth-specific protocols developed by the Bluetooth SIG. RFCOMM and the TCS binary protocol have also be developed by the Bluetooth SIG but they are based on the ETSI TS 07.10 [18] and the ITU-T Recommendation Q.931 [19], respectively. The Bluetooth Core protocols (plus the Bluetooth radio) are required by most of Bluetooth devices, while the rest of the protocols are used only as needed.

Together, the Cable Replacement layer, the Telephony Control layer, and the Adopted protocol layer form application-oriented[2] protocols enabling applications to run over the Bluetooth Core protocols. As mentioned earlier, the Bluetooth Specification is open and additional protocols (e.g., HTTP, FTP [10], etc.) can be accommodated in an interoperable fashion on top of the Bluetooth-specific transport protocols or on top of the application-oriented protocols shown in Figure 1.

[1] Not shown above OBEX in Figure 1.
[2] "Application-oriented" here is with respect to Bluetooth transport services and should be interpreted as any protocol layer, or application that runs on top of the Bluetooth-specific transport protocols.

2.1 Bluetooth Core Protocols

2.1.1 Baseband

The Baseband and Link Control layer enables the physical RF link between Bluetooth units forming a piconet [1]. As the Bluetooth RF system is a Frequency-Hopping-Spread-Spectrum system in which packets are transmitted in defined time slots on defined frequencies, this layer uses inquiry and paging procedures to synchronize the transmission hopping frequency and clock of different Bluetooth devices.

It provides 2 different kind of physical links with their corresponding baseband packets, Synchronous Connection-Oriented (SCO) and Asynchronous Connectionless (ACL) which can be transmitted in a multiplexing manner on the same RF link. ACL packets are used for data only, while the SCO packet can contain audio only or a combination of audio and data. All audio and data packets can be provided with different levels of FEC or CRC error correction and can be encrypted.

Furthermore, the different data types, including link management and control messages, are each allocated a special channel.

2.1.1.1 Audio

Audio data can be transferred between one or more Bluetooth devices, making various usage models possible and audio data in SCO packets is routed directly to and from Baseband and it does not go through L2CAP. Audio model is relatively simple within Bluetooth; any two Bluetooth devices can send and receive audio data between each other just by opening an audio link.

2.1.2 Link Manager Protocol

The link manager protocol [2] is responsible for link set-up between Bluetooth devices. This includes security aspects like authentication and encryption by generating, exchanging and checking of link and encryption keys and the control and negotiation of baseband packet sizes.

Furthermore it controls the power modes and duty cycles of the Bluetooth radio device, and the connection states of a Bluetooth unit in a piconet.

2.1.3 Logical Link Control and Adaptation Protocol

The Bluetooth logical link control and adaptation protocol (L2CAP) [3] adapts upper layer protocols over the baseband. It can be thought to work in parallel with LMP in difference that L2CAP provides services to the upper layer when the payload data is never sent at LMP messages.

L2CAP provides connection-oriented and connectionless data services to the upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions. L2CAP permits higher level protocols and applications to transmit and receive L2CAP data packets up to 64 kilobytes in length.

Although the Baseband protocol provides the SCO and ACL link types, L2CAP is defined only for ACL links and no support for SCO links is specified in Bluetooth Specification 1.0.

2.1.4 Service Discovery Protocol (SDP)

Discovery services are crucial part of the Bluetooth framework. These services provide the basis for all the usage models. Using SDP, device information, services and the characteristics of the services can be queried and after that, a connection between two or more Bluetooth devices can be established. SDP is defined in the Service Discovery Protocol specification [4].

2.2 Cable Replacement Protocol

2.2.1 RFCOMM

RFCOMM is a serial line emulation protocol and is based on ETSI 07.10 specification. This "cable replacement" protocol emulates RS-232 control and data signals over Bluetooth baseband, providing both transport capabilities for upper level services (e.g. OBEX) that use serial line as transport mechanism. RFCOMM is specified in [5].

2.3 Telephony Control Protocol

2.3.1 Telephony Control – Binary

Telephony Control protocol - Binary (TCS Binary or TCS BIN) [6], a bit-oriented protocol, defines the call control signaling for the establishment of speech and data calls between Bluetooth devices. In addition, it defines mobility management procedures for handling groups of Bluetooth TCS devices. TCS Binary is specified in the Bluetooth Telephony Control protocol Specification Binary, which is based on the ITU-T Recommendation Q.931 [19], applying the symmetrical provisions as stated in Annex D of Q.931

2.3.2 Telephony Control – AT Commands

Bluetooth SIG has defined the set of AT-commands by which a mobile phone and modem can be controlled in the multiple usage models (See Chapters 3.2 and 3.6). In Bluetooth, AT-commands, which are utilized, are based on ITU-T Recommendation V.250 [20] and ETS 300 916 (GSM 07.07) [21]. In addition, the commands used for FAX services are specified by the implementation. These may be either:

- Fax Class 1.0 TIA-578-A [22] and ITU T.31 Service Class 1.0 [23]

- Fax Class 2.0 TIA-592 [24] and ITU T.32 Service Class 2.0 [25]
- Fax Service Class 2 - No industry standard

2.4 Adopted Protocols

2.4.1 PPP

In the Bluetooth technology, PPP is designed to run over RFCOMM to accomplish point-to-point connections. PPP is the IETF Point-to-Point Protocol [10] and PPP-Networking is the means of taking IP packets to/from the PPP layer and placing them onto the LAN. Usage of PPP over Bluetooth is described in [26].

2.4.2 TCP/UDP/IP

These protocol standards are defined by the Internet Engineering Task Force and used for communication across the Internet [10]. Now considered as the most widely used protocol family in the world, TCP/IP stacks have appeared on numerous devices including printers, handheld computers, and mobile handsets. Access to these protocols is operating system independent, although traditionally realized using a socket programming interface model. The implementation of these standards in Bluetooth devices allows for communication with any other device connected to the Internet: The Bluetooth device, should it be a Bluetooth cellular handset or a data access point for example is then used as a bridge to the Internet.

TCP/IP/PPP is used for the all Internet Bridge usage scenarios in Bluetooth 1.0 and for OBEX in future versions [11]. UDP/IP/PPP is also available as transport for WAP [12].

2.4.3 OBEX Protocol

IrOBEX [27] (shortly OBEX) is a session protocol developed by the Infrared Data Association (IrDA) to exchange objects in a simple and spontaneous manner. OBEX, which provides the same basic functionality as HTTP but in a much lighter fashion, uses a client-server model and is independent of the transport mechanism and transport API, provided it realizes a reliable transport base. Along with the protocol itself, the "grammar" for OBEX conversations between devices, OBEX also provides a model for representing objects and operations. In addition, the OBEX protocol defines a folder-listing object, which is used to browse the contents of folders on remote device.

In the first phase, RFCOMM is used as sole transport layer for OBEX [11]. Future implementations are likely to support also TCP/IP as a transport.

2.4.3.1 Content Formats

vCard [13] and vCalendar [14] are open specifications developed by the versit consortium and now controlled by the Internet Mail Consortium. These

specifications define the format of an electronic business card and personal calendar entries and scheduling information, respectively. vCard and vCalendar do not define any transport mechanism but only the format under which data is transported. By adopting the vCard and vCalendar, the SIG will help further promote the exchange of personal information under these well-defined and supported formats. The vCard and vCalendar specifications are available from the Internet Mail Consortium and are being further developed by the Internet Engineering Task Force (IETF).

Other content formats, which are transferred by OBEX in Bluetooth, are vMessage and vNote [15]. These content formats are also open standards and are used to exchange messages and notes. They are defined in the IrMC specification, which also defines a format for the log files that are needed when synchronizing data between devices.

2.4.4 WAP

Hidden computing usage models can be implemented using the WAP features. Bluetooth as a WAP Bearer is defined in [12].

The Wireless Application Protocol (WAP) Forum is building a wireless protocol specification [16] that works across a variety of wide-area wireless network technologies. The goal is to bring Internet content and telephony services to digital cellular phones and other wireless terminals. In Figure 2, the protocol stack of the WAP framework is depicted.

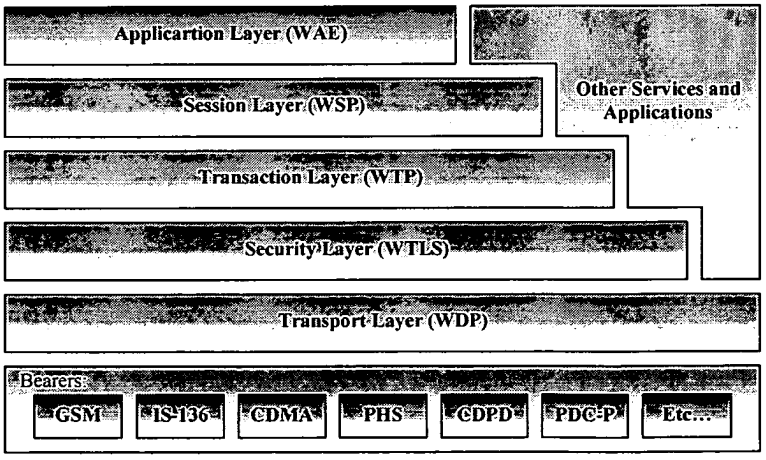


Figure 2 WAP Framework

The idea behind the choice of WAP is to reuse the upper software applications developed for the WAP Application Environment (WAE). These include WML and WTA browsers that can interact with applications on the PC. Building application gateways which mediate between WAP servers and some other application on the PC makes it possible to implement various hidden

computing functionality, like remote control, data fetching from PC to handset etc. WAP servers also allow for both content push and pull between PC and handset, bringing to life concepts like information kiosks.

WAP framework also opens up the possibility of custom applications for handsets that use WML and WML Script as "universal" Software Development Kit.

2.4.4.1 Content Formats

Supported content formats for WAP over Bluetooth are WML, WMLScript, WTA event, WBMP, and vCard/vCal. These are all part of WAE. More information on WAE can be found from [16].

3 Bluetooth Usage Models and Protocols

In this chapter, the highest priority usage models identified by the SIG's marketing group are briefly introduced. Each usage model is accompanied by a Profile. Profiles define the protocols and protocol features supporting a particular usage model. Bluetooth SIG has specified the profiles for these usage models. In addition to these profiles, there are four general profiles that are widely utilized by these usage model oriented profiles. These are the generic access profile (GAP) [28], the serial port profile [29], the service discovery application profile (SDAP) [30], and the generic object exchange profile (GOEP) [31].

3.1 File Transfer

The file transfer usage model (See also the file transfer profile [32]) offers the ability to transfer data objects from one device (e.g., PC, smart-phone, or PDA) to another. Object types include, but are not limited to, .xls, .ppt, .wav, .jpg, and .doc files, entire folders or directories or streaming media formats. Also, this usage model offers a possibility to browse the contents of the folders on a remote device.

In addition, simple push and exchange operations, e.g., business card exchange are covered in the object push profile [33], with vCard specified as the format for pushed business card content.

In Figure 3, the required protocol stack presented for this usage model is presented. The figure does not show the LMP, Baseband, and Radio layers although those are used underneath (See Figure 1).

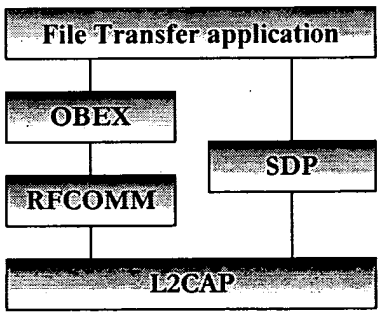


Figure 3 Protocol Stack for File Transfer Applications

3.2 Internet Bridge

In this usage model, mobile phone or cordless modem acts as modem to the PC, providing dial-up networking [8] and fax [9] capabilities without need for

physical connection to the PC. The dial-up networking scenario of this usage model needs a two-piece protocol stack (in addition to the SDP branch), which is shown in Figure 4. The AT-commands are needed to control the mobile phone or modem and another stack (E.g., PPP over RFCOMM) to transfer payload data. The fax scenario has a similar protocol stack but PPP and the networking protocols above PPP are not used and the application software sends a facsimile directly over RFCOMM.

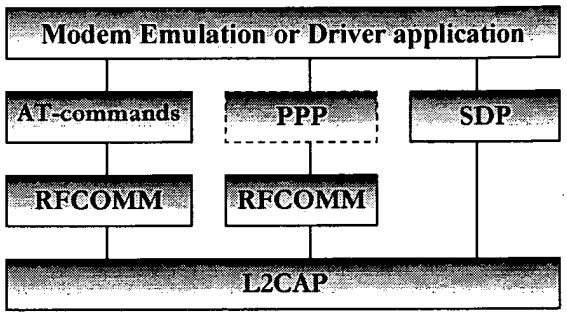


Figure 4 Dial-up Networking Protocol Stack

3.3 LAN Access

In this usage model (See also the LAN access profile [26]), multiple data terminals (DTs) use a LAN access point (LAP) as a wireless connection to a Local Area Network (LAN). Once connected, the DTs operate as if it they were connected to the LAN via dialup networking. The DT can access all of the services provided by the LAN. The protocol stack is nearly identical to the protocol stack in the Internet bridge usage model except that the AT-commands are not used. The protocol stack is represented in Figure 5.

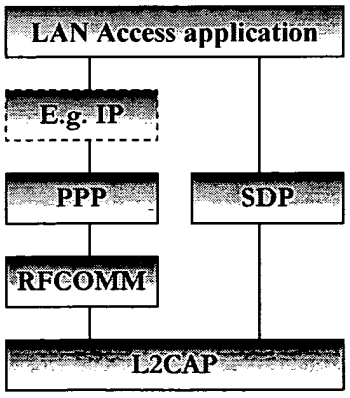


Figure 5 Protocol Stack of LAN Access (PPP) Usage Model

3.4 Synchronization

The synchronization usage model [34] provides a device-to-device (phone, PDA, computer, etc.) synchronization of the PIM (personal information management) information, typically phonebook, calendar, message, and note information. Synchronization requires business card, calendar and task information to be transferred and processed by computers, cellular phones and PDAs utilizing a common protocol and format. The protocol stack for this usage model is presented in Figure 6. In the figure, the synchronization application block represents either an IrMC client or an IrMC server software.

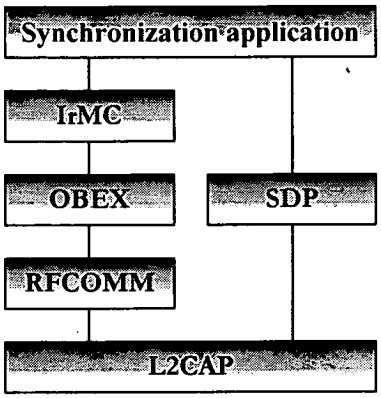


Figure 6 Protocol Stack for Synchronization

3.5 Three-in-One Phone

Telephone handsets built to this profile may connect to three different service providers. First, telephones may act as cordless phones connecting to the public switched telephone network (PSTN) at home or the office and incurring a fixed line charge. This scenario [35] includes making calls via a voice basestation, making direct calls between two terminals via the basestation and accessing supplementary services provided by an external network. Second, telephones can connect directly to other telephones for the purpose of acting as a "walkie-talkie" or handset extension. Referred to as the intercom scenario [36], the connection incurs no additional charge. Third, the telephone may act as a cellular phone connecting to the cellular infrastructure and incurring cellular charges. The cordless and intercom scenarios use the same protocol stack, which is shown in Figure 7. The audio stream is directly connected to the Baseband protocol indicated by the L2CAP bypassing audio arrow.

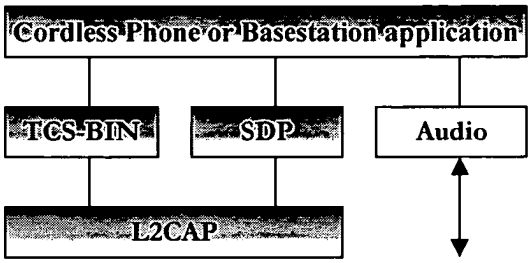


Figure 7 Protocol Stack for Cordless Phone and Intercom Scenarios

3.6 Ultimate Headset

The headset can be wirelessly connected for the purpose of acting as a remote device's audio input and output interface. The headset increases the user's freedom of movement while maintaining call privacy. A common example is a scenario where a headset is used with either a cellular handset, cordless handset, or personal computer for audio input and output. The protocol stack for this usage model is depicted in Figure 8 [7]. The audio stream is directly connected to the Baseband protocol indicated by the L2CAP bypassing audio arrow. The headset must be able to send AT-commands and receive result codes. This ability allows the headset to answer incoming calls and then terminate them without physically manipulating the telephone handset.

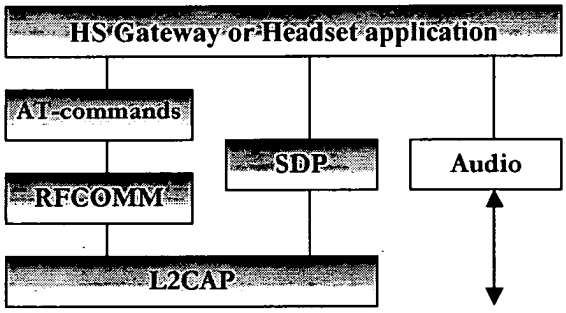


Figure 8 Ultimate Headset Protocol Stack

4 Summary

The Bluetooth protocols are intended for rapidly developing applications using the Bluetooth technology. The lower layers of the Bluetooth protocol stack are designed to provide a flexible base for further protocol development. Other protocols, such as RFCOMM, are adopted from existing protocols and these protocols are only modified slightly for the purposes of Bluetooth. The upper layer protocols are used without modifications. In this way, existing applications may be reused to work with the Bluetooth technology and the interoperability is ensured more easily.

The purpose of the Specification is to promote the development of interoperable applications targeted at the highest priority usage models identified by the SIG's marketing team. However, the Specification also services as a framework for further development. Naturally, vendors are encouraged to invent more usage models within this framework. Using the Bluetooth technology with the capabilities of current computers and communications devices, the possibilities for new future wireless applications are unlimited.

5 References

[1] Bluetooth Special Interest Group, Baseband Specification

[2] Bluetooth Special Interest Group, LMP Specification

[3] Bluetooth Special Interest Group, L2CAP Specification

[4] Bluetooth Special Interest Group, SDP Specification

[5] Bluetooth Special Interest Group, RFCOMM with TS 07.10

[6] Bluetooth Special Interest Group, Telephony Control Protocol Specification

[7] Bluetooth Special Interest Group, Headset Profile

[8] Bluetooth Special Interest Group, Dial-Up Networking Profile

[9] Bluetooth Special Interest Group, Fax Profile

[10] Internet Engineering Task Force, IETF Directory List of RFCs (http://www.ietf.org/rfc/), July 1999.

[11] Bluetooth Special Interest Group, IrDA Interoperability

[12] Bluetooth Special Interest Group, Interoperability Requirements for Bluetooth as a WAP Bearer

[13] The Internet Mail Consortium, vCard - The Electronic Business Card Exchange Format, Version 2.1, September 1996.

[14] The Internet Mail Consortium, vCalendar - The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996.

[15] Infrared Data Association, IrMC (Ir Mobile Communications) Specification, Version 1.1, February 1999.

[16] WAP Forum, WAP Forum Specifications (http://www.wapforum.org/what/technical.htm), July 1999

[17] Bluetooth Special Interest Group, Bluetooth Host Controller Interface Functional Specification

[18] ETSI, TS 07.10, Version 6.3.0

[19] International Telecommunication Union, "ITU-T Recommendation Q.931"

[20] International Telecommunication Union, "ITU-T Recommendation V.250"

[21] ETSI, TS 101 369 (GSM 07.10) version 6.1.0

[22] TIA-578-A Facsimile Digital Interface. Asyncronous Facsimile DCE Control Standard, Service Class 1

[23] International Telecommunication Union, ITU T.31 Asyncronous Facsimile DCE Control – Service Class 1.0

[24] TIA-592 Facsimile Digital Interface. Asyncronous Facsimile DCE Control Standard, Service Class 2.0

[25] International Telecommunication Union, ITU T.32 Asyncronous Facsimile DCE Control – Service Class 2.0

[26] Bluetooth Special Interest Group, LAN Access Profile using PPP

[27] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX), Version 1.2, April 1999

[28] Bluetooth Special Interest Group, Generic Access Profile

[29] Bluetooth Special Interest Group, Serial Port Profile

[30] Bluetooth Special Interest Group, Service Discovery Application Profile

[31] Bluetooth Special Interest Group, Generic Object Exchange Profile

[32] Bluetooth Special Interest Group, File Transfer Profile

[33] Bluetooth Special Interest Group, Object Push Profile

[34] Bluetooth Special Interest Group, Synchronization Profile

[35] Bluetooth Special Interest Group, Cordless Telephony Profile

[36] Bluetooth Special Interest Group, Intercom Profile

6 Acronyms

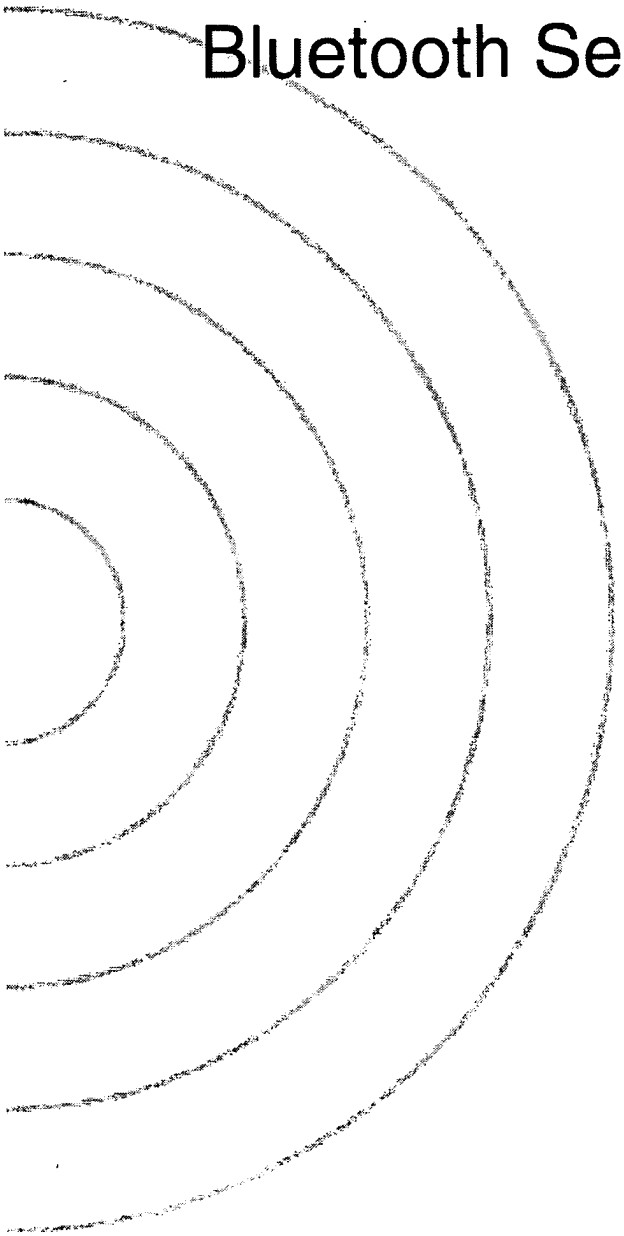
Abbreviation or Acronym	Meaning
ACL	Asynchronous ConnectionLess
API	Application Programming Interface
CRC	Cyclic Redundancy Check
DT	Data Terminal
FEC	Forward Error Correction
FTP	File Transfer Protocol
GAP	Generic Access Profile
GOEP	Generic Object Exchange Profile
HCI	Host Controller Interface
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IrDA	Infrared Data Association
IrMC	Ir Mobile Communications
LAN	Local Area Network
LAP	LAN Access Point
LMP	Link Manager Protocol
L2CAP	Logical Link and Control Adaptation Protocol
OBEX	Object Exchange Protocol
PDA	Personal Digital Assistant
PIM	Personal Information Management
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephony Network
RFCOMM	Serial Cable Emulation Protocol
SCO	Synchronous Connection-Oriented
SDAP	Service Discovery Application Profile
SDP	Service Discovery Protocol
TCP/UDP	Transport Control Protocol/User Datagram Protocol

Abbreviation or Acronym	Meaning
TCS Binary	Telephony Control Specification – Binary
WAE	Wireless Application Environment
WAP	Wireless Application Protocol
WML	Wireless Markup Language

Bluetooth WHITE PAPER	DATE 15 July 99	N.B.	DOCUMENT NO 1.C.116/1.0
RESPONSIBLE Thomas Muller	E-MAIL ADDRESS thomas.t.muller@nmp.nokia.com		STATUS

Bluetooth Security Architecture

Version 1.0



This White Paper describes a flexible security architecture for Bluetooth that allows different security levels for applications. While Bluetooth provides link-level authentication and encryption, enforcing at only this level prevents user-friendly access to more public-oriented usage models such as discovering services and exchanging business cards. This architecture uses the link-level security mechanisms of Bluetooth to enforce the service level security policy (security mode 2) of the Generic Access Profile.

Special Interest Group (SIG)

The following companies are represented in the Bluetooth Special Interest Group:
Ericsson Mobile Communications AB
IBM Corp.
Intel Corp.
Nokia Mobile Phones
Toshiba Corp.

Revision History

Revision	Date	Comments
0.0	1999-03-29	first draft, based on discussion at the SW face-to-face meeting in chandler, AZ
0.0.1	1999-03-30	Requirement on limited user intervention added 2 requirements in question added Start work on procedures (general behavior, Handling of RFCOMM)
0.0.2	1999-04-01	Incorporated feedback from Paul and Chatschik
0.0.3	1999-04-07	Feedback from Brian Redding
0.1	1999-04-09	Integrate decisions from the meeting 1999-04-08 Add interfaces of the security manager
0.2	1999-04-16	Modifications to the interfaces of the security manager: – Queries from L2CAP and other protocols harmonised – Only BD_ADDR used in query – Entity taking care of registration is implementation dependent. Registration moved to a separate section; interface to applications removed. – UI: set-up of trusted relationship included Security Policy for changed connection (section 2.1): wording changed to reflect that this includes client and server role. Section 3.1: – Pairing removed – registration can also be done by general management entity.

0.3	1999-04-27	– Remove parts for L2CAP connection hold after BB loss, because not supported by L2CAP any more: mainly changes in 3.5.2 and 3.5.3. – Flow chart changed according to phone meeting April 21st and included in document – Requirements for service security levels (requirement 3) corrected. Changes to distinguish between outgoing and incoming connections: – Default security level (in section 3.2.3) – Interface for registration: levels for both incoming and outgoing connections separately defined – Query to security manager: attribute for incoming/outgoing connection added Changes to make authentication mandatory in case authorisation is required: Statements in 3.2.1 and 3.2.3
0.5	1999-05-16	Security levels for registering multiplexing protocols added in section 3.6.5. Incorporate the changes agreed upon at the interoperability face-to-face meeting in Tampere: – Trust levels of devices might be set individually for services or groups of services. – Key management functions outside of Bluetooth mentioned – Trust flag replaced by more generic wording.
0.51	1999-05-26	Added statement on encryption in 2.1
0.8	1999-06-25	Introduction completely rewritten Requirements/Design objectives => what does the architecture provide Major editorial changes Removed chapter on consequences for Bluetooth specs Added section 4.6 Interface to HCI / Link Manager Added parameter ConnectionHandle in - 4.2 Interface to L2CAP - 4.3 Interface to other multiplexing protocols because it is needed in section 4.6 for HCI commands

0.86	1999-07-02	Incorporated changes from Chatschik and Jon • Abstraction: user ⇒ ESCE • Statement on application level security in Section 2.4 • Unknown device is also untrusted (Section 3.2.2) • Requirements for transition from security mode 2 to 3 added • Explanation for outgoing connections • Section 3.3.5.1 removed • Section 4.4: UI ⇒ ESCE and statements on calling directions
1.0	1999-07-13	Include PIN request to ESCE Terminology reference to GAP Replace initialization with bonding Editorial changes

Contributors

Paul Moran	3COM
Patric Lind	Ericsson
Patrik Olsson	Ericsson
Johannes Elg	Ericsson
Chatschik Bisdikian	IBM
Amal Shaheen	IBM
Jon Inouye	Intel
Robert Hunter	Intel
Brian Redding	Motorola
Stephane Bouet	Nokia
Thomas Müller (Owner)	Nokia
Martin Roter	Nokia

Disclaimer and copyright notice

Contents

1 Introduction

The Bluetooth specification includes security features at the link level. It supports authentication (unidirectional or mutual) and encryption. These features are based on a secret link key that is shared by a pair of devices. To generate this key a pairing procedure is used when the two devices communicate for the first time.

The link level functions are defined in the Bluetooth Baseband and the Link Manager Protocol Specifications (see [1] and [2]). More comprehensive descriptions can be found in the Bluetooth '99 conference proceedings (see [3], [4]).

The Bluetooth profiles describe how to use the Bluetooth protocols in an interoperable way. Concerning security this is done in the Generic Access Profile [5], which also defines terms used throughout this white paper. This profile specifies three security modes for a device:

- Security mode 1 (non-secure): A device will not initiate any security procedure.
- Security mode 2 (service-level enforced security): A device does not initiate security procedures before channel establishment at L2CAP level. This mode allows different and flexible access policies for applications, especially running applications with different security requirements in parallel.
- Security modes 3 (link level enforced security): A device initiates security procedures before the link set-up at the LMP level is completed.

This white paper deals with security mode 2. It describes how flexible security mechanisms can be implemented. The contents of this white paper serve only as an implementation guideline and do not represent a Bluetooth specification, since this is device specific and not required for interoperability.

Chapter 2 lists the requirements and design objectives leading to the definitions of this flexible architecture and the limitations. Chapter 3 describes a possible architecture using a central security manager.

2 Functionality & Limitations

This chapter describes the functionality of the architecture explained in Chapter 3. This can also be seen as a summary what can be built on top of security mode 2 (service-level enforced security, see [5]).

2.1 Security Levels

- It is possible to define different security levels for devices and services.
- For devices two trust levels are distinguished:
  - Trusted Device: Device with fixed relationship (paired) that is trusted and has unrestricted access to all services.
  - Untrusted Device: Device with no permanent fixed relationship (but possibly a temporary one) or device that has a fixed relationship, but is not considered as trusted. The access to services is restricted.

  A possible refinement is to set the trust level of a device specifically for services or a group of services. The interaction with the remote device does not exclude the implementation of such refined access policies.
- For services the requirement for authorisation, authentication and encryption are set independently (although some restrictions apply). The access requirements allow to define three security levels:
  - Services that require authorisation and authentication. Automatic access is only granted to trusted devices. Other devices need a manual authorisation.
  - Services that require authentication only. Authorisation is not necessary.
  - Services open to all devices; authentication is not required, no access approval required before service access is granted.
- A default security level is defined to serve the needs of legacy applications. This default policy will be used unless other settings are found in a "security" database related to a service, e.g., an internal security information database.

2.2 Flexibility and Usability

- It is possible to grant access to some services without providing access to other services (example: On a cellular phone, Service Discovery records shall be accessible, whereas dialup networking shall only be available for specific devices.)
- The security architecture supports security policies for devices with some services communicating with changing remote devices (example: File

Transfer or Business Card Exchange). Access granted to a service on such device does

  - not open up access to other services on the device.
  - not grant future access automatically or in an uncontrolled way to services on the device.

- User intervention for access to services is avoided as much as possible. It is only needed to allow devices limited access to services or for setting up trusted relationship with devices allowing unlimited access to services.
- This architecture does not deal with application level security, but such concepts are not excluded.

2.3 Implementation and Bluetooth-specific Matters

- The security architecture accounts for Bluetooth multiplexing protocols at and above L2CAP. At present, only RFCOMM is considered, as all other protocols are not Bluetooth-specific, and some have their own security features.
- The security architecture allows different protocols to enforce the security policies. For example, L2CAP will enforce the Bluetooth security policy for cordless telephony, RFCOMM will enforce the Bluetooth security policy for dialup networking, and OBEX will use its own security policy for file transfer and synchronisation.
- The architecture can completely work using security mode 2 of the Generic Access Profile. Especially since there are no changes to Baseband and LMP functions for authentication and encryption.
- Authentication and encryption are set for a physical connection (i.e., on baseband level).
- Lower layers are not aware of service/application layer security.
- The enforcement policy for authentication, authorisation or encryption might be different for client and server role. The security level of peer entities running an application needs not to be symmetric.

2.4 Risks and Limitations:

The following scenarios have been considered in identifying the limitations.

- Scenario 1: There are two Bluetooth devices (e.g., PDAs). Each device has a set of applications: calendar, file synchronization, etc. The two devices will communicate, over a Bluetooth link, to perform a certain task such as file synchronization.

- Scenario 2: There are more than two of scenario 1 devices. All devices will communicate over Bluetooth links to perform tasks that do not require security such as exchanging business cards.

- Scenario 3: A small device such as a PDA requires access, over a Bluetooth link, to infrastructure services: the Internet, e-commerce applications, corporate database, etc. Such device will be connected to a "LAN Access Point" over the BT link. The LAN Access Point will be connected to the infrastructure services via a wired or wireless LAN. This is a 3-tier configuration where tier 1 is the small device, tier 2 is the LAN Access Point, and tier 3 is the Infrastructure Services.

The Bluetooth Security architecture has the following limitations:

1. Support for legacy applications: In all scenarios, the legacy application will not make calls to the security manager. Instead a Bluetooth-aware "adapter" application is required to make security-related calls to the Bluetooth security manager on behalf of the legacy application.

2. Only a device is authenticated and not its user. If there is a need for authentication of the user, other means – e.g., application level security features – will be necessary.

3. Refer to scenario 1. There is no mechanism defined to preset authorisation per service. However, a more flexible security policy can be implemented with this architecture, without a need to change the Bluetooth protocol stack. Of course, modifications of the security manager and the registration processes would be necessary.

4. The approach only allows access control at connection set-up. The access check can be asymmetric, but once a connection is established, data flow is in principle bi-directional. It is not possible within the scope of this architecture to enforce unidirectional traffic.

5. Support for the 3-tier configuration in scenario 3: The security architecture presented in this paper is built upon the Bluetooth baseband security procedures that addresses the BT link security and mutual device authentication at each end of the link. To address the end-to-end security issue present in cases like in scenario 3, this paper assumes the existence of a "higher-level" end-to-end security solution which may utilize the Bluetooth security architecture presented for accessing devices and services directly present at the two ends of a Bluetooth link. This higher-level security solution is outside the scope of this paper.

3 Security Architecture to meet the Requirements

This chapter describes an approach for a flexible security architecture built on top of the link-level security features of Bluetooth.

3.1 Overview

The general architecture is shown in Figure 1. The key component is a security manager with the following tasks:

- Store security-related information on services
- Store security-related information on devices
- Answer access requests by protocol implementations or applications (access granted or refused)
- Enforce authentication and/or encryption before connecting to the application.
- Initiate or process input from an ESCE[1] (e.g., the device user) to set-up trusted relationships on device level.
- Initiate pairing and query PIN entry by the user. PIN entry might also be done by an application.

More details will be described in the following sections.

This approach is devoted to connection-oriented L2CAP channels. The sections up to 3.4 only deal with this. For connectionless L2CAP data transmission, restrictions apply, which will be discussed in Section 3.5.

The security architecture presented in this paper provides a very flexible security framework. This framework dictates when to involve a user (e.g., to provide a PIN) and what actions the underlying BT protocol layers follow to support the desired security check-ups. Within this framework, a number of realizations of the presented architecture can be instantiated, some of them simpler and some of them more advanced than the one discussed in detail in this paper, without moving outside the scope of the architecture.

[1] ESCE stands for "External Security Control Entity." ESCE typically represents a human operating a device who decides how to proceed with security related matters, e.g., provide a PIN whenever needed, decide to create a trust relation with a device, etc. In general though, an ESCE represents an entity with the authority and knowledge to make decisions on how to proceed in a manner consistent to this security architecture. It could be a device user, or a utility application executed on behalf of the user based on preprogrammed security policies. In the latter case, this utility could reside within or outside a particular BT-enabled device. Without lack of generality, in the sequel the terms "ESCE" and "user" will be used interchangably and without any distinction.

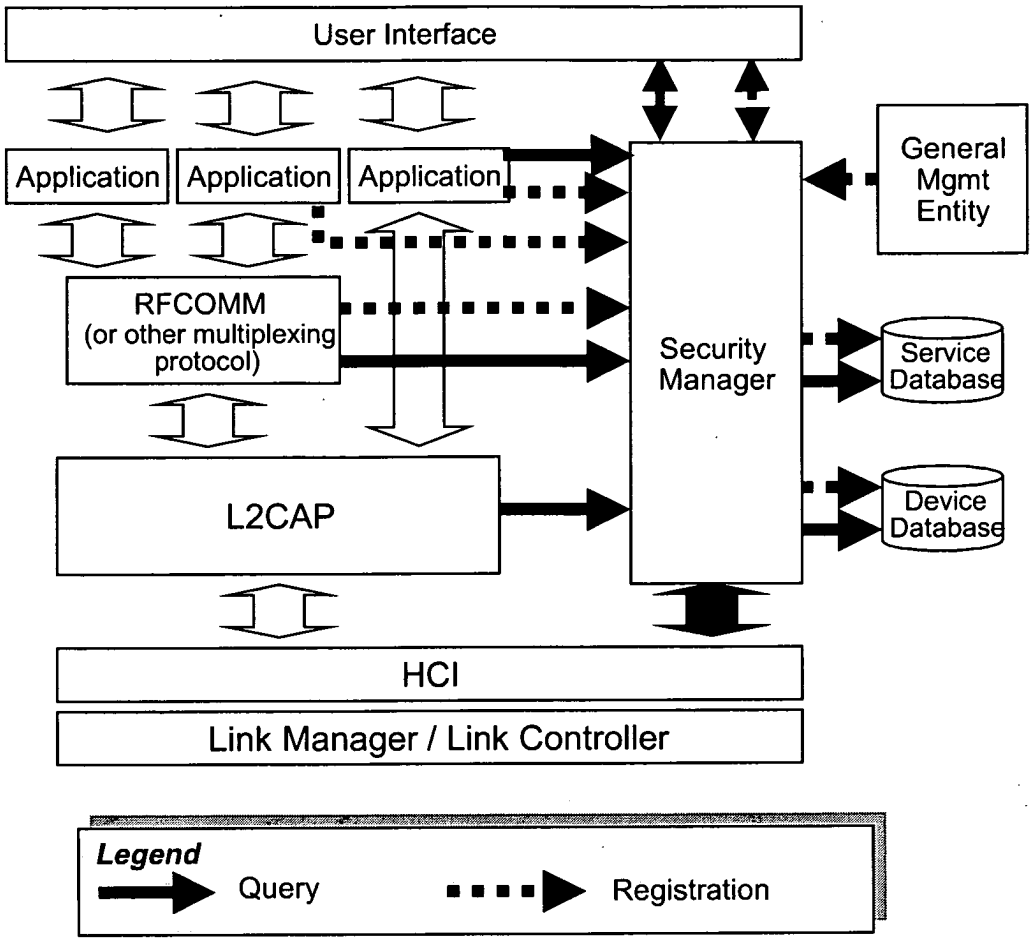


Figure 1: Security Architecture

This approach with a centralised security manager allows easy implementation of flexible access policies, because the interfaces to protocols and other entities are kept simple and are limited to query/response and registration procedures. The policies for access control are encapsulated in the security manager. Therefore, implementation of more complex policies would not affect implementation of other parts.

Implementations may decide, who performs the registration task, e.g., the application itself or a general management entity, responsible for setting the path in the protocol stack and/or registering the service at service discovery. This will be further discussed in Section 3.3.4.

3.2 Security Levels

3.2.1 Authorisation and Authentication

We distinguish between authentication and authorisation. The terms are defined as follows:

Authentication is the process of verifying 'who' is at the other end of the link. Authentication is performed for devices (BD_ADDR). In Bluetooth this is achieved by the authentication procedure based on the stored link key or by pairing (entering a PIN).

Authorisation is the process of deciding if device X is allowed to have access to service Y. This is where the concept of **'trusted'** exists. Trusted devices (authenticated and indicated as "trusted"), are allowed access to services. Untrusted or unknown devices may require authorisation based on user interaction before access to services is granted. This does not principally exclude that the authorisation might be given by an application automatically. Authorisation always includes authentication.

3.2.2 Device Trust Level

We distinguish between two different device trust levels:

Trusted Device:	The device has been previously authenticated, a link key is stored and the device is marked as "trusted" in the Device Database.
Untrusted Device:	The device has been previously authenticated, a link key is stored but the device is not marked as "trusted" in the Device Database
Unknown Device:	No security information is available for this device. This is also an untrusted device.

There will be a database table maintained in the security manager (see below). This database might be maintained for all services together (normal case referred to throughout this paper) or separately for each service or group of services.

3.2.2.1 Authenticate trusted device

The verification is done using the authentication procedure, defined in the LMP and Baseband specifications. A device is verified as trusted, if a positive authentication response is given and the trusted flag is set.

3.2.2.2 Set-up of the trusted relationship

A trusted relationship is established during the pairing procedure. This is usually performed during the bonding procedure but could be performed at connection set-up.

When an untrusted device is authorised to use a service, it is also possible to add it to the list of trusted devices during the same procedure. This of course requires an active selection by the user.

3.2.2.3 Authenticate untrusted device

Authentication of untrusted devices is done similarly as for trusted devices with the exception that the device is not marked as trusted in the internal database.

3.2.3 Security Level of Services

The security level of a service is defined by three attributes:

Authorisation Required:	Access is only granted automatically to trusted devices (i.e., devices marked as such in the device database) or untrusted devices after an authorisation procedure. Authorisation always requires authentication to verify that the remote device is the right one.
Authentication Required:	Before connecting to the application, the remote device must be authenticated
Encryption Required:	The link must be changed to encrypted mode, before access to the service is possible

This information is stored in the service database of the security manager.

If no registration has taken place, a default security level is used. This default is:

Incoming Connection:	Authorisation and Authentication required
Outgoing Connection:	Authentication required

3.3 Connection Set-up Procedure

3.3.1 General Concept

To meet different requirements on availability of services without user intervention, we must perform the authentication after determining what the security level of the requested service is. Thus, the authentication cannot be performed, when the ACL link is established. The authentication is performed, when a connection request to a service is submitted.

Figure 2 illustrates the information flow for access to a trusted service. This is intended to be an example to understand and discuss the basic principles. The details will be described further below.

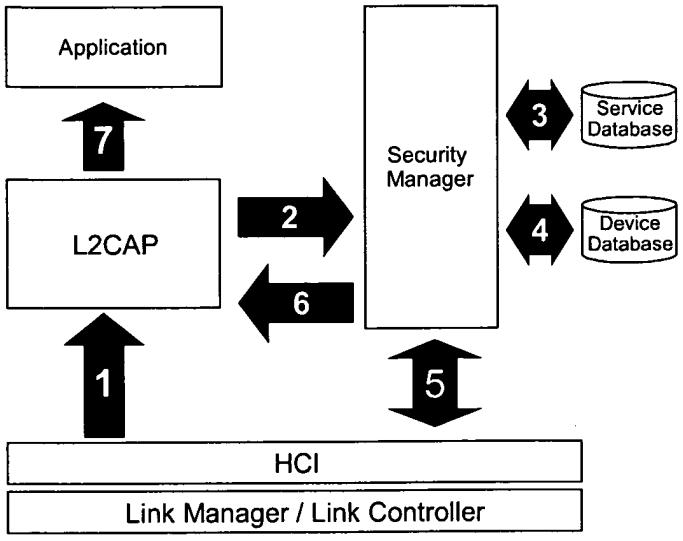


Figure 2: Information flow for access to trusted service

The following procedures are performed:

1. Connect request to L2CAP
2. L2CAP requests access from the security manager.
3. Security manager: lookup in service database
4. Security manager: lookup in device database
5. If necessary, security manager enforces authentication and encryption
6. Security manager grants access
7. L2CAP continues to set-up the connection.

Authentication can be performed in both directions, client authenticates server and vice versa.

3.3.2 Authentication on Baseband Link Set-up

Although not targeted to security mode 3 (link level enforced security, see [5]), this architecture can support this mode as well. The security manager can command the link manager to enforce authentication before accepting a BB connection using the HCI. Different modes could be implemented in parallel:

- authentication on BB connection
- authentication on connection to the applications

Clearly, they cannot run simultaneously, so if both are implemented, a decision has to be made somewhere. Before transitioning from mode 2 to mode 3, it must be ensured that untrusted devices do not get unwanted

access. To achieve this, the security manager can remove any link keys for untrusted devices stored in the radio module. The security manager can use the HCI.

3.3.3 Handling for Protocol Stack

For incoming connections, the access control procedure is described in Figure 3 for incoming connections. Access control is required at L2CAP and in some cases additionally at multiplexing protocols above (e.g., RFCOMM). When receiving a connection request, the protocol entity queries the security manager, providing any multiplexing information it received with the connect request. The security manager makes a decision whether access is granted or refused and replies to the protocol entity. If access is granted, the connection set-up procedure is continued. If access is refused, the connection is terminated.

If no access control is performed on a protocol layer, no interaction takes place with the security manager or other entities.

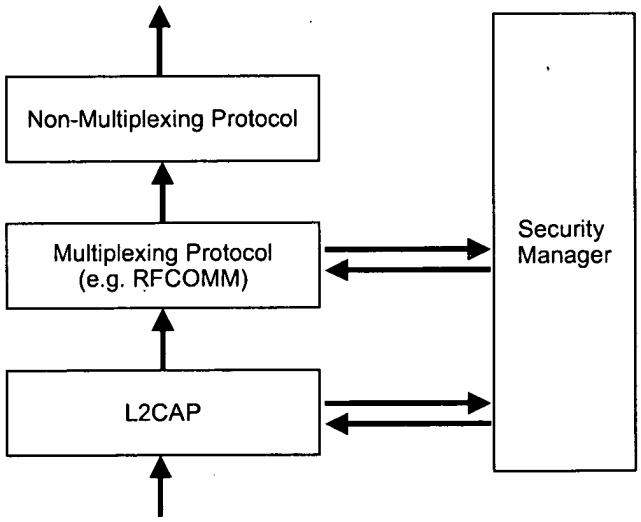


Figure 3: Behaviour of Protocols for incoming Connections

In this model we have two queries (e.g., function calls) to the security manager. The security manager should be able to store information on existing authentications . This avoids multiple authentication procedures on LMP level (i.e., over the air) within the same session.

Thus, RFCOMM will do a policy check call to security manager. This will require an additional function call but not necessarily an additional authentication.

For outgoing connections, a security check might also be required to achieve mutual authentication (authorisation is probably not useful here in most

cases). A similar procedure is carried out. The most elegant way is of course, if the applications submit requests to the security manager directly. If this is not possible (e.g., legacy applications), queries to the security manager are submitted by all multiplexing protocols from top to bottom, see Figure 4.

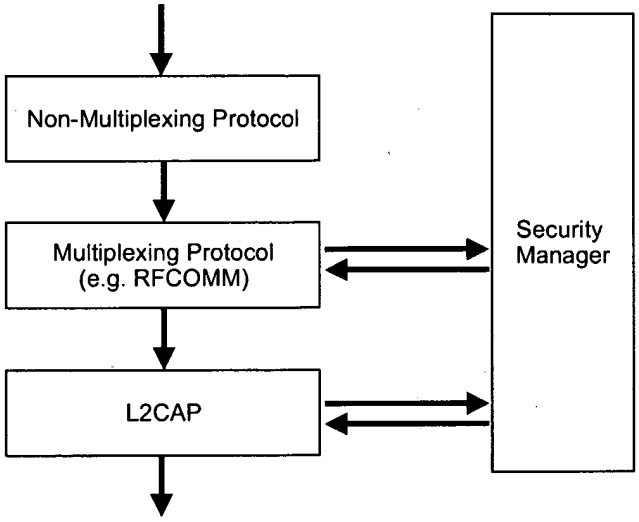


Figure 4: Behaviour of Protocols for outgoing Connections

3.3.4 Registration Procedures

As mentioned in section 3.1, the security manager maintains security information for services in security databases, the implementation of which is outside the scope of this paper. Applications must register with the security manager before becoming accessible, see Figure 5.

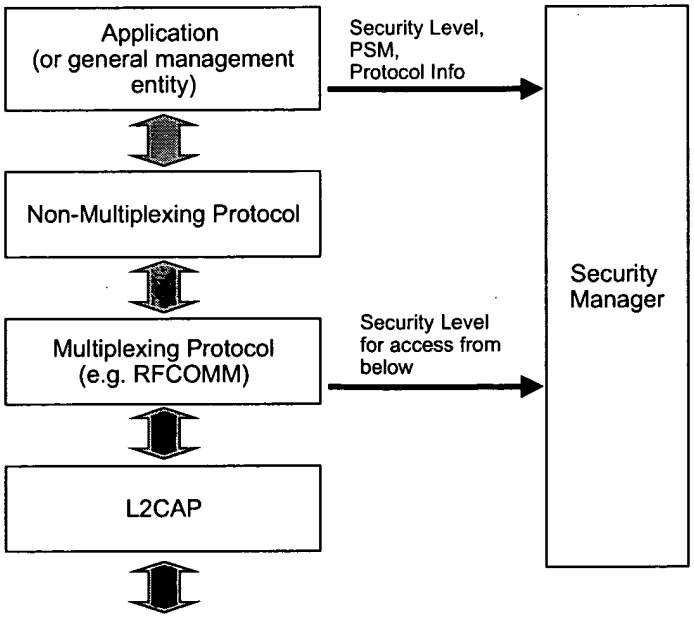


Figure 5: Registration Processes

Applications or their security delegate must provide their security level and multiplexing information (This is somewhat similar to the information registered to service discovery. The latter is required for the security manager to make a decision on a request submitted by a protocol entity as this entity will not know about the final application in most cases).

Multiplexing protocol implementations performing queries at the security manager should register the policy for accessing them from below.

Both registrations can also be done by the entity that is responsible for setting the path in the BT protocol stack. It is implementation-dependent, which entity does the registration (note: combining the service and security registration processes is not excluded).

If no registration has taken place, the security manager will assume the default security level to make a decision on access see Section 3.2.3.

L2CAP does not require registration here. It is the first multiplexing protocol in the Bluetooth stack and there will be a query for every connection request.

3.3.5External Key Management

This architecture does not exclude use of external key management procedures. Key management applications can distribute PINs or the link keys directly.[2] In this case though, one needs to proceed with caution to maintain proper interoperability of the BT-enabled devices.

3.4 Access Control Procedures

This section gives an example how the access control can be handled in the security manager. Other solutions with the same functionality and/or the same security level are possible.

[2] It is not possible to provide the encrpytion keys from outside the module.

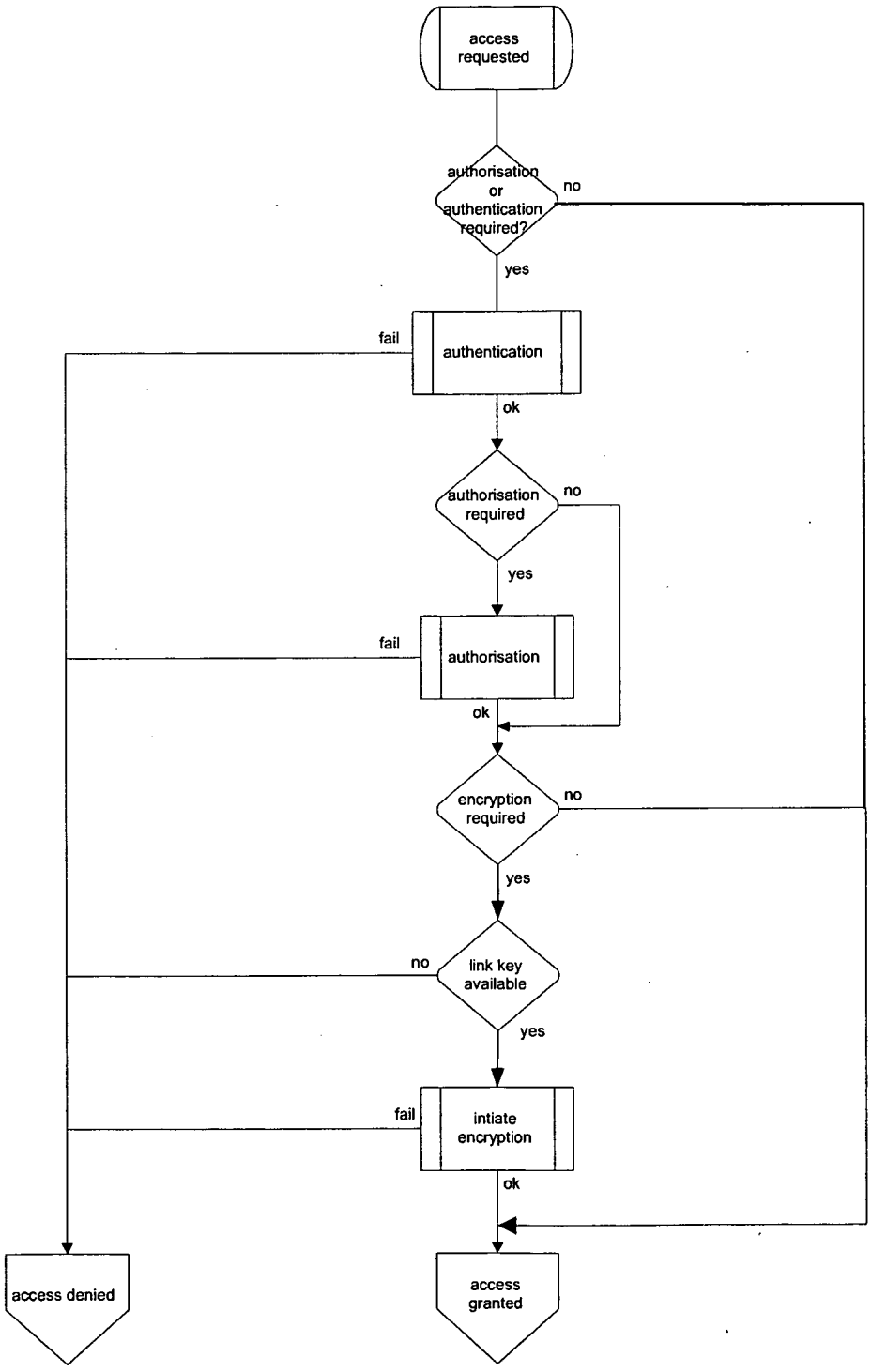


Figure 6: Example Flow Chart for Access Check by the security manager

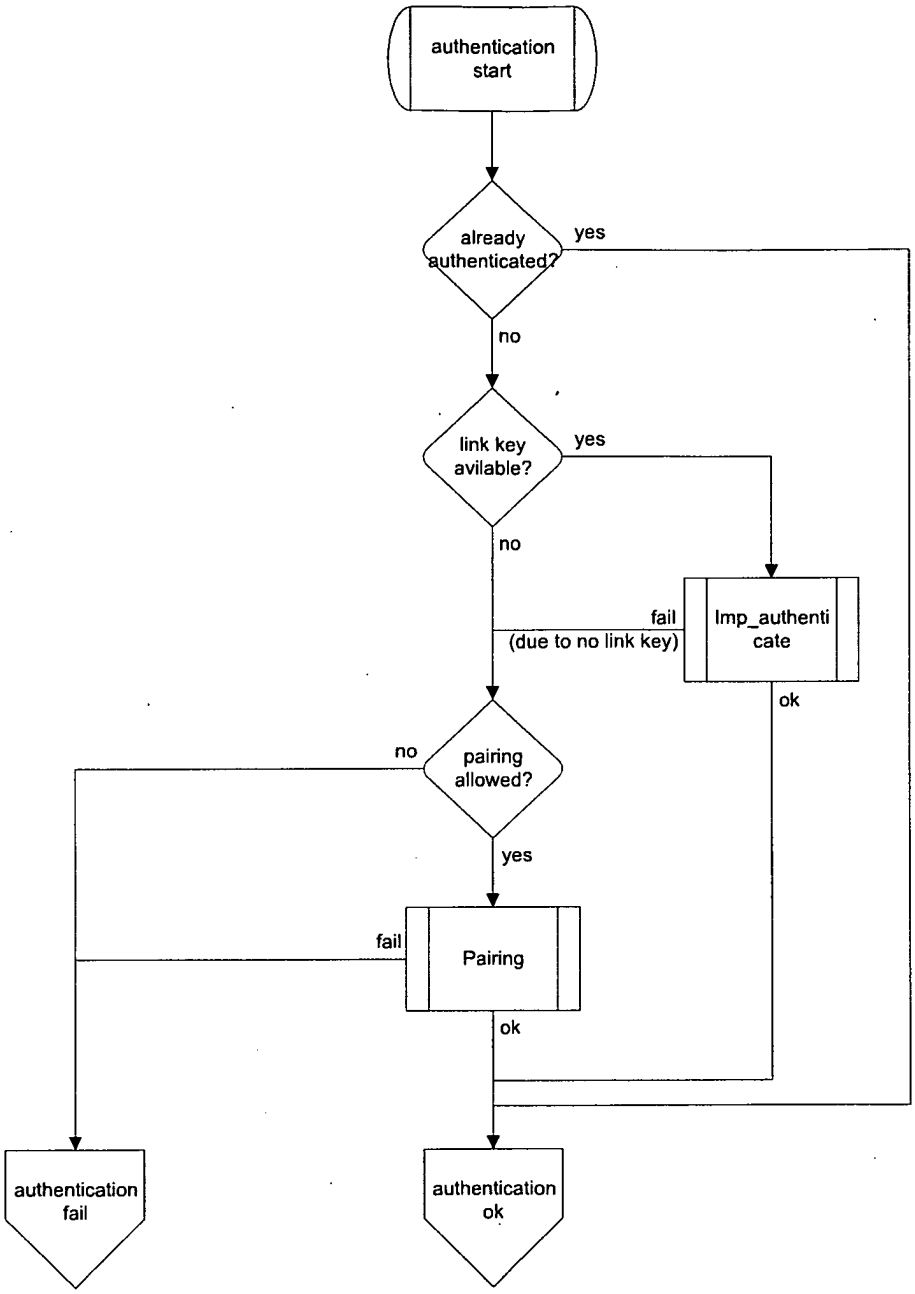


Figure 7: Example Flow Chart for Authentication Procedure

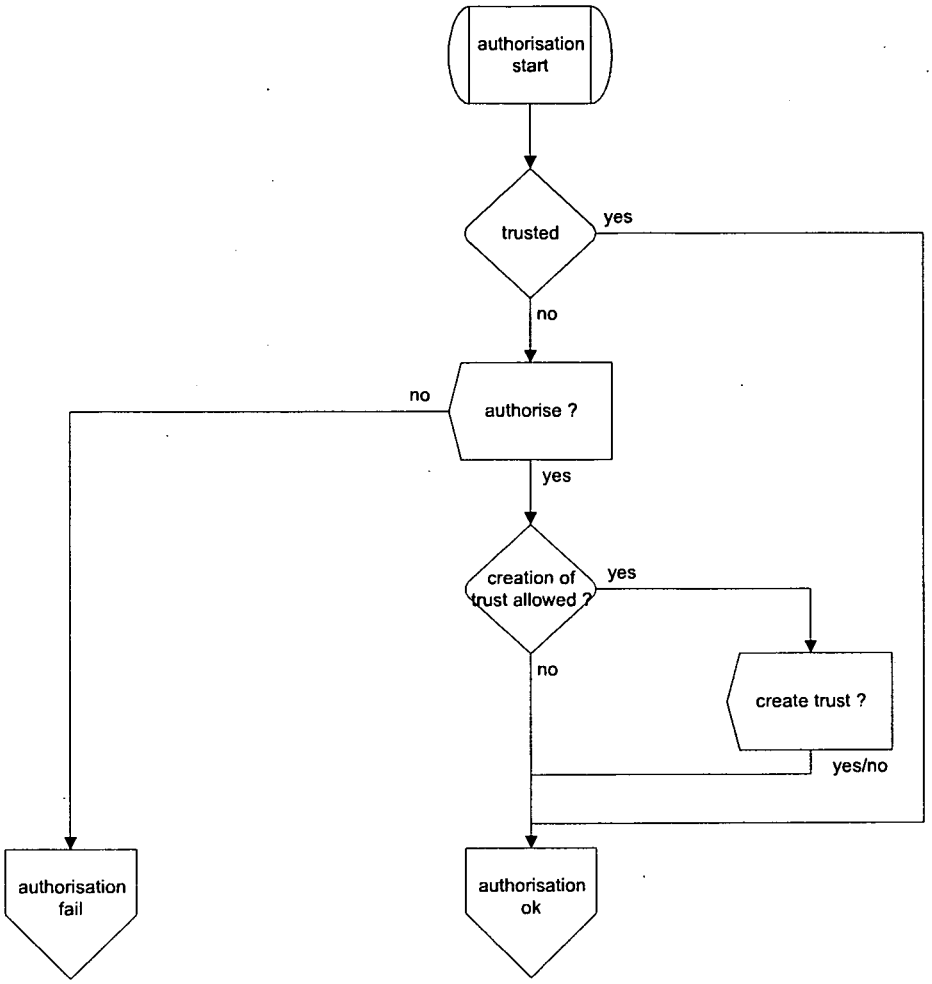


Figure 8: Example Flow Chart for Authorisation Procedure

3.5 Connectionless L2CAP

As the security check is performed at a connection request to L2CAP to set-up a connection to the next higher protocol or application, the security check of connectionless data packets cannot be performed in that way. It is not practical to perform a security check on each single connectionless data packet. Therefore, a general policy of handling connectionless packets has to be made at L2CAP level.

L2CAP offers the possibility to block connectionless traffic. This block can be done for a single protocol (PSM) on top of L2CAP, a list or all protocols. The same choices are possible for enabling connectionless traffic.

The security manager should check, whether there is any service in the service data base that does not permit connectionless data packets. The security manager will then initiate the enabling/disabling accordingly.

If connectionless packets are passed, there will be no security check. It is then up to the protocol above L2CAP to make sure that no unacceptable security problem occurs. It will always be known, whether the data came in via connectionless or connection-oriented mode, but for connectionless packets the originiator is not known or verifiable.

4 Interfaces and Functions of the Security Manager

This chapter describes a possible set of interfaces and functions for the security manager. The interactions are modelled as function calls. This chapter is meant as an implementation example. Clearly, the internal interfaces have no impact on interoperability with remote devices.

4.1 Databases

The security manager that implements the security architecture in this paper has to maintain several databases (or in general "information lists").

For the database tables the following abbreviations are used:

- "M" for mandatory to support
- "O" for optional to support
- "C" for conditional to support

The statements mandatory, optional or conditional are relative to the discussed realization of the security architecture. Clearly, simpler or more elaborate realizations of the security implementation may have other mandatory, conditional, or optional entires.

4.1.1 Service Database

The service database has to maintain the following security-related entries for each service. This could be stored in non-volatile memory or the services register at start-up.

Authorisation Required:	M	Boolean
Authentication Required:	M	Boolean
Encryption Required:	M	Boolean
PSM Value	M	Uint16
Broadcasting allowed	O	Boolean
other routing information	C	Structure TBD (only when needed)

4.1.2 Device Database

The trust information has to be stored in non-volatile memory. If entries are deleted for some reason, the respective device is treated an unknown, i.e., set to the default access level as defined in 3.2.3.

BD_ADDR	M	48 Bit IEEE address

trust level	M	Trusted / untrusted
link key	M	Bit field (length up to 128 bit)
device name	O	String (can be used to avoid name request)

Note: It would also possible to store key information in a different way.

4.1.3 Temporary Storage

There are some data that should be stored to reduce overhead on the air interface. For each Baseband link:

- Authentication status
- Encryption status

4.2 Interface to L2CAP

L2CAP asks the security manager for access rights to a service on incoming and outgoing connection requests. There is no registration procedure since L2CAP is mandatory in Bluetooth protocol stacks.

access := SEC_accessRequest (
ProtocolIdentification,
ChannelIdentification,
BD_ADDR,
ConnectionHandle
IncomingConnection
);

Parameter	I/O	Type	Content
access	ret[3]	boolean	true = granted false = denied
ProtocolIdentification	in	uint32	Number assigned to identify, which protocol has submitted the query The value is set to zero in case of L2CAP.
PSM	in	uint32	The channel is identified by the PSM value of next protocol in

[3] "ret" stands for the value "returned" by a function call; "in" stands for the "input" parameters to a function call; "out" stands for additional parameters that are "outputed" when the function call returns.

			stack
BD_ADDR	in	48 bit	48 bit address of the remote device
ConnectionHandle[4]	in	uint16	connection handle (on HCI level) associated with ACL link to remote device
IncomingConnection	in	boolean	true = incoming connection false = outgoing connection

4.3 Interface to other multiplexing Protocols

Other multiplexing protocols (e.g., RFCOMM) that need to make decisions on access to services query the security manager in a similar way as L2CAP. There is an additional registration procedure, which allows to set the access policy for connection to the multiplexing protocol itself.

access := SEC_accessRequestMultiplexingProtocol (
ProtocolIdentification,
ChannelIdentification,
BD_ADDR,
ConnectionHandle
IncomingConnection);

Parameter	I/O	Type	Content
access	ret	boolean	true = granted false = denied
ProtocolIdentification	in	uint32	Number assigned to identify, which protocol has submitted the query
ChannelIdentification	in	uint32	Channel ID (or whatever is used in that protocol), where a decision of an access policy is based on; for RFCOMM: DLCI
BD_ADDR	in	48 bit	48 bit address of the remotes device
ConnectionHandle[5]	in	uint16	connection handle (on HCI level) associated with ACL link

[4] The information is redundant to the Bluetooth device address. However, the connection handle is needed to initiate authentication and encryption via the HCI.

			to remote device
IncomingConnection	in	boolean	true = incoming connection false = outgoing connection

4.4 Interface to ESCE (e.g., UI)

The architecture includes user interaction for authorisation purposes. This includes access permission to services and setting up a trusted relationship to a remote device.

access = SEC_authorisationRequest (
ServiceName,
DeviceName,
&FutureTrustedRelationship = false
);

The security manager calls the ESCE (e.g., the user interface); incoming parameters are the information submitted with the request, outgoing parameters hold the response.

Parameter	I/O	Type	Content
ServiceName	in	String	human readable name of the application (from registration of application)
DeviceName	in	String	human readable name of the device (retrieved using the name request or from internal database)
FutureTrustedRelationship	out	Boolean	If this value is true, the remote device will be marked as trusted. Default value is false.

If a PIN is requested by the security manager, the following call to the ESCE can be used. The PIN entry can also be requested directly from the link manager (then the security manager requests authentication, and the link manager performs the necessary actions if no valid link is available.

SEC_PinRequest (

[5] See footnote in previous section.

BD_ADDR,
Name,
PIN
);

Parameter	I/O	Type	Content
BD_ADDR	in	48 Bit	48 bit address of the remote device
Name	in	String	Bluetooth device name (human readable name)
PIN	out	String	Bit field, length < 16 bytes

In case the the ESCE wants the security manager to create a trusted relationship outside of other procedures, a simple command may be used:

SEC_createTrustedRelationship (
BD_ADDR
);

The ESCE (e.g., user interface) calls the security manager.

Parameter	I/O	Type	Content
BD_ADDR	in	48 bit	48 bit address of the remote device

4.5 Registration Procedures

There are certain registration procedures necessary:

- services with their security level and protocol stack information
- multiplexing protocols above L2CAP

This registration can be done by the entity that is responsible for setting the path in the BT protocol stack. It is implementation-dependent, which entity does the registration. Without registration the default settings apply.

SEC_registerApplication (
Name,
SecurityLevel,
PSM,
ProtocolIdentification,

ChannelIdentification
);

Parameter	I/O	Type	Content
Name	in	string	human readable name of the application (intended for user queries)
SecurityLevel	in	uint16	Bit 0–2 incoming connection: bit 0 = authorisation required bit 1 = authentication required bit 2 = encryption required Bit 3–5 outgoing connection: bit 3 = authorisation required bit 4 = authentication required bit 5 = encryption required Bit 6 = reception of connectionless packets allowed
PSM	in	uint16	PSM value used at L2CAP level
ProtocolIdentification	in	uint32	Number assigned to identify, which protocol has to make the decision for access. Zero = decision at L2CAP
ChannelIdentification	in	uint32	Channel ID (or other appropriate multiplexing identifier), where a decision of an access policy is based on; for RFCOMM: DLCI In case of *ProtocolIdentification =0*, this value has to be ignored.

SEC_registerMultiplexingProtocol (
ProtocolIdentification
LowerProtocol,
LowerChannel,
Security Level
);

Parameter	I/O	Type	Content
ProtocolIdentification	in	uint32	Number assigned to identify, which protocol is registered

LowerProtocol	in	uint32	ProtocolIdentification of the next lower protocol
LowerChannel	in	uint32	ChannelIdentification used at the lower layer
Security Level	in	uint16	Bit 0–2 incoming connection: bit 0 = authorisation required bit 1 = authentication required bit 2 = encryption required Bit 3–5 outgoing connection: bit 3 = authorisation required bit 4 = authentication required bit 5 = encryption required Bit 6 = reception of connectionless packets allowed

4.6 Interface to HCI / Link Manager

4.6.1.1Authentication request

For requesting an authentication of a remote device, the HCI_Authentication_Requested command and as an answer the Authentication Complete event are used which are shown below. For further details please refer to 4.5.15 and 5.2.6 of [6]

Command	OCF	Command Parameters	Return Parameters
HCI_Authentication_ Requested	0x0011	Connection_Handle	

Event	Event Code	Event Parameters
Authentication Complete	0x06	Status, Connection_Handle

4.6.1.2 Encryption control

For encryption control, the HCI_Set_Connection_Encryption command and as an answer the Encryption Change event are used to enable and disable the link level encryption. For further details please refer to 4.5.16 and 5.2.8 of [6].

Command	OCF	Command Parameters	Return Parameters
HCI_Set_ Connection_ Encryption	0x0013	Connection_Handle, Encryption_ Enable	

Event	Event Code	Event Parameters
Encryption Change	0x08	Status, Connection_Handle, Encryption_Enable

4.6.1.3 Name request to remote device

For a name request to a remote device, the HCI_Remote_Name_Request command and as an answer the Remote Name Request Complete event are used which are shown below. For further details please refer to 4.5.19 and 5.2.7 of [6].

Command	OCF	Command Parameters	Return Parameters
HCI_Remote_Name_ Request	0x0019	BD_ADDR, Page_Scan_Repetition_ Mode, Page_Scan_Mode, Clock_Offset	

Event	Event Code	Event Parameters
Remote Name Request Complete	0x07	Status, BD_ADDR, Remote_Name

4.6.1.4 Set encryption policy at link level

The general encryption policy at link level can be set by the HCI_Write_Encryption_Mode command which will be answered by the Command Complete event, both shown below. The Encryption Mode parameter controls if the Bluetooth radio will require encryption at link level for each connection with other Bluetooth radios. For further details please refer to 4.7.25 and 5.2.14 of [6].

Command	OCF	Command Parameters	Return Parameters
HCI_Write_ Encryption _Mode	0x0022	Encryption_ Mode	Status

Event	Event Code	Event Parameters
Command Complete	0x0E	Num_HCI_Command_Packets, Command_Opcode, Return_Parameters

4.6.1.5 Set authentication policy at link level

The general authentication policy at link level can be set the HCI_Write_Authentication_Enable command which is shown below. The Authentication_Enable parameter controls if the Bluetooth radio will require authentication at link level for each connection with other Bluetooth radios. As for the HCI_Write_Encryption_Mode answer, the Command Complete event is used. For further details please refer to 4.7.23 and 5.2.14 of [6].

Command	OCF	Command Parameters	Return Parameters
HCI_Write_ Authentication _Enable	0x0020	Authentication_Enable	Status

5 References

[1] Bluetooth Baseband. Bluetooth Specification Section B:14

[2] Bluetooth Link Manager Protocol, Bluetooth Specification Section C

[3] Thomas Müller: Bluetooth Security. Proceedings Bluetooth'99, London, June 1999

[4] Joakim Persson: Bluetooth Baseband Security Concept. Proceedings Bluetooth'99, London, June 1999

[5] Generic Access Profile, Bluetooth Specification Section

[6] Bluetooth Host Controller Interface, Bluetooth Specification Section H1

THE IMMINENT MARKET EXPLOSION OF BLUETOOTH WIRELESS INTERFACES ON PRODUCTS SIGNALS AN RF MEASUREMENT FIRST FOR MANY TEST ENGINEERS.

On your marks for testing Bluetooth

THE BLUETOOTH WIRELESS STANDARD is coming into its own, and hundreds of millions of Bluetooth-enabled products will ship by the end of 2002. The Bluetooth technology will be self-contained within many products; for others, it will be an addition in the form of a PC Card that plugs into a mobile device or a dongle that plugs into a desktop system's RS-232 or parallel-printer port. As Bluetooth becomes ubiquitous, you'll find yourself having to test Bluetooth devices at the protocol-stack and RF levels.

A complete Bluetooth module comprises a radio transceiver, a baseband-link controller, and a link manager (**Figure 1**). The baseband-link controller connects the radio hardware to the baseband-processing and physical protocol layer. The link manager performs high-level protocol activities such as link setup, authentication, and configuration. Application-layer software sits above the link manager.

A Bluetooth implementation can operate as a single-chip (integrating radio and protocol stack) or multi-chip (separate radio and protocol ICs) design. Cellular-phone manufacturers may wish to combine the Bluetooth baseband function in the same device as the GSM (Global Standard for Mobile Communication) baseband function and, therefore, will want a separate Bluetooth radio. A digital camera manufacturer is more likely to select a single-chip design to simplify assembly.

RF, PROTOCOL, AND PROFILES

Three aspects of a Bluetooth module need testing: RF, protocol, and profiles. You can perform many of the RF measurements with standard test instruments such as spectrum analyzers with vector demodulation, transmitter analyzers, power meters, digital-signal generators, and BERTs (bit-error-rate testers).

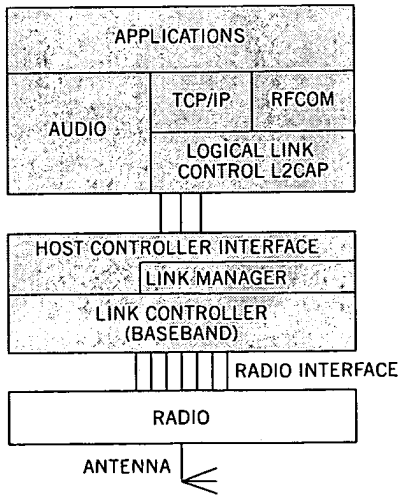
Some of the measurements, however, require the radio to form a standard Bluetooth radio-link connection with the test instrument and the test instrument to have some control over the equipment under test (**Figure 2**). For these tests, the test system must be able to support the Bluetooth protocol to make the link. Consequently, you can expect instrument vendors to develop new test instruments similar to the integrated radio test sets for digital cellular radio.

For protocol tests, you can use protocol sniffers that monitor and display data moving between Bluetooth devices. You can also use products such as the EBDK (Ericsson Bluetooth Development Kit) (**Reference 1**). Ericsson will soon release a version of the EBDK, known as a Blue Unit, that will include software for use during early qualification testing. Eventually, some companies should develop complete reference test systems.

A *profile* is the application level protocol that makes a device perform its functions as the user expects. All Bluetooth devices that claim to offer a given functionality must use the appropriate profile from the Bluetooth specification. This requirement ensures interoperability between common devices from multiple vendors.

For example, the LAN access profile defines a data connection between a DT (data terminal) and a LAP

Figure 1



A complete Bluetooth module comprises a radio transceiver, a baseband-link controller, and a link manager.

This article appeared in Test & Measurement World/ September 2000.

(LAN access point). The profile defines the following services and connection states for the application layer: initialization of LAN access service, shutdown of LAN access service, establishment of LAN connection, loss of LAN connection, and disconnection of LAN connection.

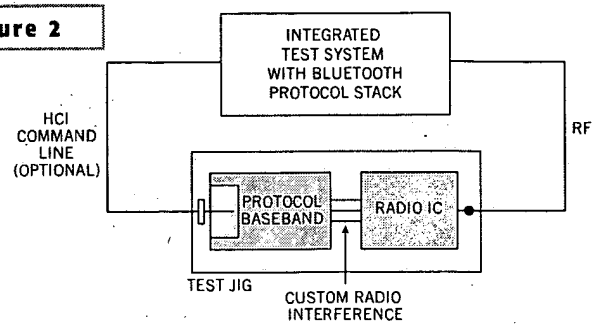
Until a reference test system is available, early Bluetooth profile testing will require product-to-product interoperability testing. To facilitate this, a series of "Unplugfests" have been arranged by the Bluetooth Special Interest Group (www.bluetooth.com). At these meetings, companies with functional products can test product interoperability against products from other suppliers.

RF AND SINGLE-CHIP MEASUREMENTS

The Bluetooth radio specification outlines the performance requirements for the radio and the test to confirm conformance (**Table 1**). To measure the performance of a Bluetooth module, the test instrumentation must be able to establish a Bluetooth link with the EUT (equipment under test). It can then put the EUT into test mode. Test mode is a mandatory feature of a Bluetooth module in which the EUT can enter a loopback mode or can disable frequency hopping for making BER measurements, for example. A Bluetooth test system should also be able to disable hopping, set specific frequencies for tests, and control the transmit power level.

During design and development, you'll want to test the radio in isolation from the protocol stack that controls it. In these cases, you need to control the radio so you can set frequencies and levels to make raw transceiver measurements. You can then feed the radio output directly into a spectrum analyzer with vector signal analysis or into a transmitter analyzer and power meter to make the measurements (**Figure 3**). The measurements you can make will vary according to the control that the radio IC manufacturer provides over the IC connectors. The radio will have inputs controlling data in and out, Tx on, clocks, and supply voltage. Without a protocol stack, the radio may not function at all.

Figure 2



Some Bluetooth tests require the test instrument to support the Bluetooth protocol to form a standard Bluetooth radio-link connection.

Many radio designs permit you to feed PRBS (pseudo-random bit sequences) into the transmitter modulator and use manufacturer-specific control lines to force the radio to transmit continuously at one frequency. Doing so enables you to make frequency, power, and modulation measurements as well as output spectrum measurements. As an alternative method to radio IC testing, you can build a test jig that holds the radio IC and has a protocol stack built onto it. This method allows for comprehensive testing and simulates the integrated module approach outlined below.

There is no standard for the interface between a protocol stack and a radio, typically known as the radio interface. You must use a test fixture to hold the radio IC and provide it with some baseband control to give Bluetooth functionality. This arrangement lets you test radio ICs with a Bluetooth link and baseband control.

The connection between the radio and the test instrument vary depending upon the Bluetooth device implementation. Some radios or integrated modules have printed antennas as part of the design. In this case, you may only be able to make a connection over the air to an antenna on the test instrument input. If you use this approach, then you must characterize the path loss for each of the 79 Bluetooth frequencies. If the radio IC has an RF output connector, a direct connection to the test instrument simplifies calibrated power and sensitivity measurements. Even if you use a direct connection, though, you should measure and correct for the path loss at each frequency.

TESTING OEM BLUETOOTH PRODUCTS

OEMs buying commercial Bluetooth chip sets still need to test. Inevitably, packaging can influence the performance of a finished product because of the antenna's position as well as other internal electronics. In a mobile phone, the other

HOW TO QUALIFY PRODUCTS WITH BLUETOOTH INTERFACES

No product may be sold as "Bluetooth enabled" without first demonstrating compliance with the Bluetooth specification. You can refer to the rules laid out in the Bluetooth Qualification Program when you check compliance. Qualification is essential for ensuring that consumers have a good experience with Bluetooth-branded products. You must make sure that interoperability between products supplied by different manufacturers is guaranteed.

To obtain qualification, a manufacturer must first become a Bluetooth member by signing the adopters agreement. Two additional bodies help qualify a product: a BQTF (Bluetooth Qualification Test Facility) and a BQB (Bluetooth Qualification Body).

A BQTF is an accredited organization with the skills and equipment to test a product based on the Bluetooth specification. A BQTF may choose not to offer qualification for every aspect of the Bluetooth standard. For example, many profiles are limited to a few specific applications, and some aspects of the Bluetooth specification are optional. The BQTF performs measurements on behalf of the manufacturer on the appropriate radio, protocol, and profiles for the equipment under test. The BQTF prepares a test report that forms part of a compliance folder that is submitted to the BQB.

The role of the BQB is to review all submitted documentation and ensure that all the appropriate tests have been performed and passed satisfactorily. If all is well, the product is listed as Bluetooth qualified and may be sold as Bluetooth enabled.

electronics will, by definition, include an interfering radio transceiver. Similarly, PCs have high-speed clocks or noisy buses that can degrade module sensitivity.

In the OEM production environment, the tests have to validate performance in the shortest possible time. Production engineers need to select the subset of the conformance test specifications that are appropriate for their products' requirements.

To confirm that the device will operate

TABLE 1–THE BLUETOOTH RADIO SPECIFICATION INCLUDES CERTAIN RADIO TESTS TO CONFIRM CONFORMANCE

Transmitter tests

Test	Limits	Hopping	Test mode	Loopback (or Tx)	Payload	Packets	Certification script
Output power	20 dBm, 4 dBm, 0 dBm	On	On	Loopback	PRBS 9	Longest supported	Low/middle/high frequencies over one packet
Power control	Power class dependent +20 to −30 dBm	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over one packet
Modulation characteristics	140 kHz ≤Δf ≤175 kHz Δf≥ 115 kHz Δf2/Df1 ≥ 0.8	Off	On	Loopback	10101010 & 11110000	Longest supported	Low/middle/high frequencies over 10 packets
Initial frequency	±75 kHz	Off and on	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over 10 packets
Frequency drift	DH 1 ±25 kHz DH 3 ±40 kHz DH 5 ±40 kHz Overall: ≤ 4000 Hz/ 10 µs	Off and on	Tx	Loopback or Tx test	1010...	Longest supported	Hopping off low/middle/high over 10 packets, all supported lengths. Hopping on all frequencies over 10 packets, all supported lengths
Power density	<20 dBm per 100 kHz EIRP	On	On	Loopback	PRBS 9	Longest supported	One minimum peak power
TX output spectrum frequency range	Country-dependent (e.g., Euro/US 2.4 to 2.4835 GHz)	Off	On	Loopback	PRBS 9	DH 1	−80 dBm/Hz EIRP
TX output spectrum 20 dBm	$\Delta f = \lvert f_h - f_l \rvert$ ≤1.1 MHz	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies
TX output spectrum adjacent channel	≤−20 dBm ACP other conditions apply	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies with conditions
Out of band spurious	30 MHz to 12.5 GHz	Off	On	Loopback	PRBS 9	DH 1	Maximum power, high low frequency

Receiver tests

Test	Limits	Hopping	Test mode	Loopback (or Tx)	Payload	Packets	Certification script
Sensitivity single-slot	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over 1,600,000 returned payload bits. Dirty transmitter
Sensitivity multi-slot	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 5 (or DH 3 if DH 5 not supported)	Low/middle/high frequencies over 1,600,000 returned payload bits. Dirty transmitter
Maximum input	−20 dBm at receiver input	Off	On	Loopback	PRBS 9	DH 1	Low/middle/high frequencies over 1,600,000 returned payload bits
C/I performance	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Two generators required: BT and BT modulated
Blocking performance	BER≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Two generators required: BT and CW
Intermodulation	BE≤0.1%	Off	On	Loopback	PRBS 9	DH 1	Three generators required: BT, BT modulated, and CW

Notes: ACP (adjacent channel power); BT (Bluetooth); DH (Data High); EIRP (equivalent isotropically radiated power); PRBS (pseudo-random bit sequence); TX (transmitter).

over the Bluetooth specification's 10-m range, engineers will still need to measure sensitivity and power levels. The conformance specification requires engineers to measure receiver sensitivity as a BER of more than 1,600,000 bits at three frequencies. This test alone would take at least 25 sec using standard single-slot DH1 packets and so, in practice, the test will measure fewer bits even at a reduced number of frequencies.

In addition to RF measurements, OEMs should perform a functional test. In the case of a Bluetooth-enabled digital camera, for example, functional testing can include sending an instruction over the Bluetooth interface to activate a shutter release with flash. Engineers would need to create this command at a high level in the protocol stack (typically the application level), and the test equipment would need to package the command into the Bluetooth format. Validating the camera's response to a high-level command would give the manufacturer confidence that the interface was functioning correctly, although it is not necessarily a guarantee of robust performance.□

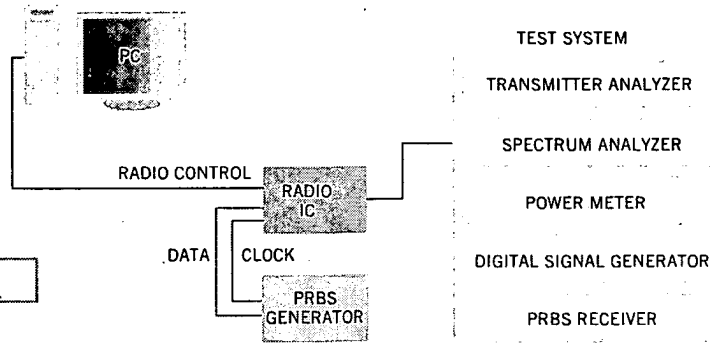


Figure 3

In development, you make raw transceiver measurements of a Bluetooth radio IC in isolation from the protocol stack.

REFERENCE

1. Available from Symbionics. www.symbionics.co.uk/solutions/bluetooth/Bluetoothkit.shtml.

AUTHOR BIOGRAPHY

Angus Robinson is product marketing manager for RF and microwave test instruments with Anritsu, Stevenage, UK. He joined Anritsu in 1998 having previously worked with Marconi Instruments (now IFR) after receiving a Bachelor of Engineering degree in electronics at Liverpool University in 1982.

BACKGROUND TO BLUETOOTH

As a standard for wireless communication between multiple devices, Bluetooth supports voice and data. In 1994, Ericsson began to develop the standard at its Lund site in Sweden (Table A). The project was initially called MC Link (multicommunicator link). In 1997, Ericsson approached other companies with a mutual interest in defining an open standard for a wireless link. In February 1998, five promoters—Ericsson, IBM, Intel, Nokia, and Toshiba—formed the SIG (Special Interest Group) to promote the standard, which they renamed Bluetooth.

The Bluetooth SIG announced the standard on May 20, 1998. Two years later, more than 1800 companies had joined the consortium as adopters of the technology. The consortium's objective is to create a de facto short-range wireless communication standard that all companies could use. In the autumn of 1999, the promoter group was expanded to nine companies, adding 3Com, Lucent Technologies, Microsoft, and Motorola. Although today the Bluetooth SIG standard is "owned" by the promoter group, it is expected that the standard will become an IEEE standard (802.15) this year and remain fundamentally the same.

Bluetooth offers wireless communication between one or more devices over a 10-m range with a maximum gross data rate of 721 kbps in the unlicensed 2.4-GHz ISM band. The purpose of the technology is to offer a low-cost, simple-to-use alternative to wired connections. As such, the potential user base is large and varied.

The first adopters of Bluetooth are expected to be mobile- phone manufacturers with wireless headsets. Mobile phones could also interface with a Bluetooth-enabled PCs to exchange files or e-mail. A PC could have a wireless mouse and keyboard, and office printers could become Bluetooth enabled so that wireless printing is possible in any location. Other early adopters of the technology include PDAs (personal digital assistants), LAN access points, digital cameras, and security-access cards.

TABLE A—GENERAL BLUETOOTH RADIO SPECIFICATION

Parameter	Specification	Comments
Link type	SCO (Synchronous connection oriented)	Point-to-point, full duplex link, circuit switched, symmetric with fixed-interval slot reservation
	ACL (Asynchronous connectionless link)	Momentary connection, packet switching asynchronous with polling access
Frequency	ISM band, 2.402 to 2.480 GHz, 79 channels—1-MHz spacing, hopping at 1600 hops/s	Now a common frequency plan including France, Spain, and Japan
Modulation	2 FSK with 0.5 bandwidth time Gaussian filter, peak deviation 175 kHz, modulation index 0.28 to 0.35	
Data rate	1 Mbaud gross, maximum net data rate 721 kbps, voice channel 64 kbps	Actual data rate depends upon packet length, use of error correction, and encryption
Transmitter power	Class 1, P >0 dBm, Class 2, P −6 to +4 dBm, Class 3, P <0 dBm	

ONLINE feature

Bluetooth Bites Information Retrieval

Maryellen Mott Allen

ONLINE, May 2001

Internet world is replete with buzzwords, trends, and rumors about the latest technologies coming to the fore. Ask anyone these days what the new darling of the Internet is, and they will tell you without a doubt that it's wireless technologies. Online journals, newsgroups, discussion forums, and print publications concerning the Internet are engaged in a furious exchange of opinions in which the pros and cons of various wireless data transmission protocols are alternately praised and reviled. One such protocol, simultaneously adored and despised, is Bluetooth (http://www.bluetooth.com).

According to Cahners In-Stat Group (http://www.instat.com) in its July 2000 report, "Bluetooth 2000: To Enable the Star Trek Generation," the manufacture of Bluetooth-enabled equipment will exceed one billion units by 2005, and the market will be worth some $5 billion. Frost & Sullivan (http://www.frost.com) is equally optimistic. It forecasts global shipments of Bluetooth-enabled products to reach over 11 million units in 2001 and predicts $2.5 billion in revenues.

The name itself is enough to start you wondering. When discussing the dry topic of data transmission protocols, we are used to throwing around terms such as TCP/IP, RIP, or PPP. Whether or not we can decipher the acronym, the terms have been used so often that most people have some idea of what we are talking about. But mention Bluetooth to the uninitiated and you're likely to be met with puzzled looks.

A BIT ABOUT BLUETOOTH

Bluetooth, named after the 10th century Viking king Harald Bluetooth, is a de facto data transmission standard developed by Ericsson and backed by a host of other technology companies that make up the Bluetooth Special Interest Group (including Ericsson, Intel, Puma Technology, Microsoft, Motorola, Nokia, 3Com, Lucent Technologies, and Toshiba). Bluetooth employs the unlicensed radio frequency (RF) portion of the electromagnetic spectrum in the 2.4-2.4835 gHz range. More simply, Bluetooth is a low-power, spread-spectrum technology that uses frequency-hopping to ensure speedy, short-range wireless data transfer of up to 720 kbps at a range of 30-100 feet. Intended as a replacement for the short-range data cable, Bluetooth's original conception was as a wireless peripheral interface, linking print- ers, monitors, keyboards, and other peripheral devices to the CPU. However, the potential for the technology is much greater, offering the ability to link devices of different types, forming instant workgroups across multiple platforms in a seamless fashion. So how does Bluetooth work?

The Bluetooth standard is a complex conglomeration of protocols arranged in a protocol stack, which, when diagramed, somewhat resembles a Dagwood-style sandwich. To put it more simply,

Bluetooth relies upon radio frequencies to transmit data, providing a universal bridge between devices on a network, or between devices from the outside and an existing network using a combination of circuit and packet-switching technologies. Bridges have the ability to link together different types of networks because a bridge delivers data based upon the MAC (Medium Access Control) address that is hard-coded into the network hardware by the factory that manufactures it, and it is unique for every device. This differs from other network devices, such as routers, that deliver network data based upon routable protocols such as TCP/IP. Because bridges operate at a lower level of the protocol stack, standards like Bluetooth can link unlike devices within a local network. The data is, like all network traffic, divided into packets. However, because Bluetooth is designed to work in the RF environment–a very noisy portion of the electromagnetic spectrum–it employs the use of shorter packets for transmission, and combines this with fast frequency-hopping to ensure a fairly robust connection.

When a message is sent over a Bluetooth connection, each packet is transmitted on a different frequency within a range of 2.4-2.4835 gHz. After the first packet is transmitted, the Bluetooth controller hops to a different frequency before sending out the next packet. The determination of which frequency within the given range will be used to start packet transmission is semi-random and is controlled by the Bluetooth Radio portion of the protocol stack. This frequency remains fixed for the duration of each packet. However, once the initial transmission frequency is chosen for the first packet, the remaining frequencies are cycled through for each subsequent packet in a determined fashion. This is known as the phase.

BLUETOOTH IS SOCIAL

When two or more Bluetooth-enabled devices are within range of each other, they automatically start communicating. Each Bluetooth-enabled device periodically broadcasts an inquiry message to see if there are any other Bluetooth devices in the area. If there is a response from another device, the originator of the inquiry message becomes the Master unit and the responder becomes the Slave. This arrangement is completely dependent upon which device is the first to send the inquiry message out when it comes within range, and does not resemble a client/server relationship. After this initial contact, the Master sends the slave(s) information about how they will communicate (i.e. the initial frequency and phase). These ad hoc networks are called Piconets. Collections of Piconets form Scatternets. Bluetooth supports point-to-point as well as point-to-multipoint links depending upon the number of devices within communication range. In each case, the connections are peer-to-peer.

Bluetooth can send data along four different channels simultaneously. Included is an asynchronous data channel (meaning packets are not sent out in pre-selected timeslots governed by the processor clock, but rather whenever there's an opportunity), and up to three simultaneous synchronous voice channels, or a combination channel, simultaneously supporting asynchronous data and synchronous voice. Each synchronous voice channel can support transfer rates of 64 kbps, and the asynchronous channel can support a bi-directional (two-way) asymmetric link of up to 721 kbps in either direction while permitting 57.6 kbps in the return direction.

So what does all of this have to do with libraries and information professionals? A lot, if the push to promote the standard is successful.

WILL LIBRARIES DEVELOP A TASTE FOR BLUETOOTH?

Imagine an environment where users enter the library with their personal digital assistants (PDAs)

and are instantly connected to the network. They can pull up their records, search proprietary databases and the Web, check email, download information, and even check out materials from the stacks without ever having to stop at the checkout desk. All of these things are theoretically possible using the Bluetooth standard. Bluetooth-enabled networks have the potential to allow libraries to offer the kind of value-added services often held up as the holy grail of mission statements.

In Ken Varnum's insightful article entitled "Information at Your Fingertips: Porting Library Services to the PDA," appearing in the September/ October 2000 issue of ONLINE, he illustrates how Ford Motor Company's corporate library is taking the initiative in porting library services to handheld devices, such as the PDA. The article demonstrates precisely how the engaged and aware information professional is finding new avenues for reaching out to users. Varnum explains how the library has effectively employed a third-party software package to translate Web pages into PDA-friendly files that users can download and take along with them on trips away from the office. This technology allows users to view documents, check and respond to email messages, and even fill out online forms. The drawback, of course, is that the user must return to the office and interface the PDA with a PC in order to synchronize the devices and actually transmit the email, form, etc. The PDA itself is not connected online and does not allow for real-time operation.

Now, take that scenario and replace the old PDA with a new Bluetooth-enabled PDA. Then suppose traveling users find themselves at a hotel, airport, or library with a Bluetooth-enabled network connected to the Internet. Our users can now not only respond to email messages, but send them as well. They can conduct research and download updated documents. They can even use their PDAs to make phone calls. In short, the user has been given the type of information access that would have been unimaginable just a few years ago.

BLUETOOTH IN TRADITIONAL SETTINGS

Even the more traditional academic or public libraries could reap the benefits of a Bluetooth-enabled network. Librarians at the reference desk assisting patrons in their research could directly beam the results to the patron's PDA, avoiding the cost of printing out the documents. Patrons with Bluetooth enabled devices connected to the library network could work together in ad hoc user groups, sharing information electronically. Library instructional sessions could be greatly facilitated by turning a patron's personal device into an instant workstation, eliminating the need for the library to purchase and maintain expensive computer labs. Network printers could be placed throughout the library for those who wish to obtain printouts without the restrictions that come with traditional network cables or the line-of-sight problems that accompany devices using infrared. A prototype description of Swedish rail traffic in the future (http://www.swedetrack.se/usblue2.htm) proposes placing Bluetooth units at public libraries close to train stations to be used as passenger interfaces.

LEXIS-NEXIS has teamed up with a primary developer of the Bluetooth Standard and member of the Bluetooth SIG, Red-M, a subsidiary of the Dutch company Madge Networks, to launch m-news. With content from LEXIS-NEXIS, m-news will provide subscribers with email updates and hypertext links to stories located on the Red-M Web site that detail Bluetooth industry developments. Registration is required but the service is free (http://www.redm.com/aboutred/newsroom/content.asp?Article=21).

And if all of that sounds too good to be true, you may be right.

BLUETOOTH MAY HAVE PROBLEMS AT ITS ROOT

The potential for Bluetooth is enormous, but its road to acceptance has been rife with criticisms and obstacles, some of which are substantial indeed. First there is the issue of cost. The original model developed by the Bluetooth SIG in 1998 called for a figure of $5 to integrate Bluetooth radio transceivers and link-level controllers into hand-held devices. The reality has been more in the range of $50 for the components, or slightly less if purchased in large quantities. And this is the price paid by the original equipment manufacturers (OEMs). Undoubtedly, by the time the device reaches the market, the price will be considerably higher. This has effectively put it out of the reach of most consumers. The inflated prices, however, are not the result of corporate greed, but more the unpredictable consequence of the technical difficulty of producing the chips and the resulting size of the chip being somewhat larger than anticipated.

Another significant problem is that of interference. As Joe Wilcox, staff writer for CNET News.com notes in his September 15 article, "As Bluetooth Nibbles, Competition Lurks" (http://news.cnet.com/news/0-1003-200-2784702.html), the issue of frequency interference could potentially harm Bluetooth a great deal. Because Bluetooth uses the unlicensed radio portion of the electromagnetic spectrum, transmissions in that band must compete with industrial microwave ovens, stadium lights, garage door openers, cordless telephones, and virtually any wireless household appliance, even baby monitors. More detrimental to the long-term health of the Bluetooth standard, however, is its interference with two other wireless standards using the same 2.4 gHz band that are older and more established: that of the 802.11B and the HomeRF. While these standards lack the catchy name and media hype that Bluetooth currently enjoys, 802.11B allows portable electronic devices to connect to an existing network at distances up to 300 feet rather than the paltry 100 feet that limits Bluetooth. Additionally, HomeRF recently won a ruling from the FCC that allows it to expand its bandwidth to 5mHz (up from 1mHz), effectively increasing transmission speeds to 10 Mbps. While there is room for these standards to operate in a complementary fashion within a wireless network environment, another challenge rears its head: compatibility.

The bickering over who has control over various portions of the spectrum gets even uglier when dealing with the unlicensed portion that Bluetooth and other standards exploit. There is little that the FCC can do to regulate this area and still preserve the integrity of maintaining an open portion of bandwidth. To get around some of the interference problems, both HomeRF and Bluetooth use frequency-hopping. In contrast, the 802.11B standard has developed around a direct-sequence model in which only one frequency is used for transmission. The use of these two different methods introduces incompatibility between devices. A frequency-hopping Bluetooth device and a direct-sequence 802.11B device would not be able to communicate with each other.

Add to that the security problems inherent in the Bluetooth standard, and the outlook appears grim. There is a security layer within the specification for Bluetooth, but by all accounts, it is easily misunderstood and prone to confusion. As Wilcox reports, an analyst with Gartner, a leading Internet industry consulting firm states, "Bluetooth is a disaster waiting to happenÉthe specs cover (security), but unless you know what you're doing, it's possible to implement the spec in such a fashion (that) you aren't doing anything worthwhile."

IT TAKES TWO TO TANGO

The rollout of Bluetooth-enabled devices has experienced many delays as a result of some of these issues. When the Bluetooth SIG was first formed, the company fully expected to see widespread implementation by Christmas of 2000. Unfortunately, it has been only recently that Bluetooth-equipped devices have begun to trickle out of the development labs and into the hands of

consumers. Even so, there are not enough Bluetooth products around to make it a worthwhile investment. One Bluetooth device is no good–you really need at least two to play. Yet it is important to bear in mind that these problems are not insurmountable. Dozens of companies have significant resources sunk into the development of Bluetooth. To abandon it would represent a tremendous loss of time and money. While the future of the technology is by no means secure (there are a lot of kinks to work out), you can rest assured that much effort will go into its continued development and promotion. We'll see within the next few years if Bluetooth will sink its teeth into the information industry, or bite itself in the butt.

Harald Bluetooth

Harald Bluetooth, son of Denmark's first king, Gorm the Old, from whom the present Danish queen, Margarethe II,,traces direct descent, did not have blue teeth. Instead, the name refers to a great man with a dark complexion. The Danish word for blue, *blå,* also meant dark and the words for man, mand, and tooth, *tand,* sound much the same. Harald Bluetooth is credited with Christianizing Denmark, Norway, and parts of Sweden and with uniting the countries into one kingdom. At the time of his rule, somewhere between 940 and 980 AD, southern Sweden was part of Denmark. In southern Sweden is the city of Lund, which is where Ericsson developed the Bluetooth technology. According to Ericsson in its discussion of Harald Bluetooth (http://bluetooth.ericsson.se/bluetoothf/beginnersg/default.asp?page=2), "One of his skills was to make people talk to each other....," hence the choice of Bluetooth to name this communications standard. If you're ever in Jelling, Denmark, you can view the rune stone Harald Bluetooth raised in honor of his parents. There's another, vastly newer, rune stone dedicated to Bluetooth himself. It's outside Ericsson's Mobile Communications office in Lund.

Maryellen Mott Allen *(mallen@lib.usf.edu) is Instructor Librarian, University of South Florida Tampa Campus Library.*

Comments? Email letters to the Editor at marydee@infotoday.com.

[infotoday.com] [*ONLINE*] [Current Issue] [Subscriptions] [Top]

Wireless Access to Internet via Bluetooth: Performance Evaluation of the EDC Scheduling Algorithm

Raffaele Bruno, Marco Conti, Enrico Gregori
Consiglio Nazionale delle Ricerche
Istituto CNUCE
Via G. Moruzzi, 1, 56124 Pisa – Italy
Tel: (0039) 050-3153063, Fax: (0039) 050-3138091
e-mail: {Marco.Conti,Enrico.Gregori}@cnuce.cnr.it
Raffaele.Bruno@guest.cnuce.cnr.it

Abstract
Bluetooth is an emerging technology for constructing ad-hoc wireless Personal Area Networks (WPANs). In this paper we analyze an innovative scheduling algorithm for asynchronous data traffic specifically tailored to the Bluetooth characteristic. This algorithm, named Efficient Double-Cycle (EDC), dynamically adapts the polling frequency to the traffic conditions. By considering a scenario where a Bluetooth master is used as wireless access point to the Internet, we show that our EDC scheduler significantly enhances the system performance with regard to a Round Robin (RR) scheduler.

Keywords
Bluetooth, TCP, Scheduling, Medium Access Control (MAC), Polling, Automatic Repeat Request (ARQ).

1. Introduction
The technologies for WPANs, as the emerging Bluetooth technology, offer a wide space for innovative solutions and applications that could bring to a radical change in everyday life. In particular, the Bluetooth technology is much more than a wireless connection for a nomadic access to the Internet or for cable replacement, but it wants to be an enabling technology for the global mobility of devices and services [2], [4].

Before Bluetooth applications can be deployed, it is necessary a considerable effort from the research community to resolve the technical issues specific to this technology. In particular, the strong need for low-cost, low-power and low-complexity devices has lead the Bluetooth standardization forum to adopt a centralized Time Division Duplex (TDD) access scheme as the MAC protocol for the channel access. In a Bluetooth network, named as *piconet*, one station has the role of master and all other Bluetooth stations are slaves. Up to seven slaves can participate to the piconet communications. The master decides which slave is the one that can access the channel. More precisely, a slave is authorized to deliver a single packet to the master only if it has received a polling message from the master.

Hence the scheduling algorithm is a key component in a Bluetooth network, the standard documents initially proposes a Round Robin (*RR*) scheduler [1]. Previous works [6], [4] have shown that the RR algorithm may introduce bandwidth wastage. Therefore, efficient MAC scheduling algorithms need to be designed. The original contribution of this paper is the performance evaluation of an efficient scheduling algorithm named Efficient Double-Cycle (*EDC*), considering Internet traffic. Our algorithm tunes the polling order to the network traffic conditions to limit the channel-bandwidth wastage caused by the polling of empty stations.

Johansson et al. have proposed a Fair Exhaustive Polling (*FEP*) [6]. Similarly to our algorithm, FEP tries to avoid the polling of inactive stations. Specifically, they define the meaning of active and inactive state for the slaves, and introduce polling sub cycles where only the active slaves are polled in a round robin fashion. Our algorithm further extends this idea i) by separating the scheduling of the uplink and downlink transmissions, and ii) by adopting a truncated binary exponential-backoff algorithm to determine the amount of time an inactive slave is removed from the polling cycle.

The design of a scheduler suitable for the Bluetooth MAC layer was also considered by Shorey et al. They have proposed several MAC scheduling algorithms [8], [9] where they distinguish the master-slave pairs based on the state of the queues at the master and slaves. They assume that the information regarding the status of the queue at the slave is available in the master because it comes directly from the slaves, which appropriately set some bits in the MAC header. Once that different classes/priorities are assigned to each master-slave pair, various service policies can be devised. Our scheduling algorithm departs from this approach because it does not rely upon any information coming from slaves.

In [10] we have exhaustively studied the EDC behavior from the MAC layer standpoint. Specifically, we have investigated several performance figures, as aggregate link utilization, MAC delays and power indexes, and we have

This work was carried out in the framework of the CNUCE-CNR GMD-FOKUS bilateral project "SIMTO – A Framework for a Flexible SIMulator Toolset for Future IP Networks ".

1st Workshop on Wireless Mobile Internet 7/01 Rome, Italy
© 2001 ACM ISBN 1-58113-423-1/01/07...$5.00

found that EDC always outperforms RR. This analysis was a theoretical one, as we do not consider the real traffic and the impact of high-layer protocols on the traffic spacing. Since one of the most interesting Bluetooth applications is expected to be the ubiquitous wireless access to Internet, in this work we extend the previous analysis to a realistic case in which the impact of TCP flow-control mechanisms has been investigated.

The performances of TCP over Bluetooth have been investigated by others. A preliminary analysis of TCP performance over Bluetooth is presented in [7], but this analysis is only related to the TCP Vegas that is rarely adopted in current TCP implementation. In [9] a more accurate analysis of TCP performances has been executed, but no insight is given on the impact of TCP parameters over the performances of a connection delivered via a Bluetooth link.

The results presented in the following indicate that EDC provides a significantly throughput improvement for the TCP connections, when compared with the RR scheduler. Particular attention has been devoted to the study of the role of the TCP Maximum Segment Size (MSS) over the throughput performance. We also investigate the impact of channel errors over the TCP and the effectiveness of the ARQ scheme adopted in Bluetooth for error recovery.

The paper is organized as follows. In Section 2 we briefly describes the Bluetooth MAC layer, with a particular attention to the channel access scheme. In Section 3 we give the specification of our scheduling algorithm. Then, Section 4 presents a complete performance evaluation of the scheduling algorithm. Concluding remarks are summarized in Section 5.

2. Overview of the Bluetooth MAC Protocol

From a logical standpoint, Bluetooth belongs to the contention-free token-based multi-access networks [4]. A Time Division Duplex (TDD) scheme of transmission is adopted. The channel is divided into time slots, each 625 μs in length. The time slots are numbered according to the Bluetooth clock of the master. The master can begin a new transmission in even numbered time slots. Odd numbered time slots are reserved for the beginning of slaves' transmissions. As stated before, the channel access is managed according to a polling scheme. The master decides which slave is the only one to have the access to the channel by sending to him a packet. The master packet may contain data or can simply be a polling packet. When the slave receives a packet from the master it is obliged to transmit in the next time slot, to acknowledge the master transmission. This implies that the scheduling decisions are always related to a couple of stations: the master and the slave that receives the master transmission. The slave packet can contain data or can simply be a NULL packet.

There are two types of physical links that can be established between Bluetooth devices: a Synchronous Connection-Oriented (*SCO*) link, and an Asynchronous ConnectionLess (*ACL*) link. The first type of physical link is a point-to-point, symmetric connection between the master and a specific slave. It is used to deliver delay-sensitive traffic, mainly voice. The SCO link rate is 64 kbps and it is settled by reserving a couple of consecutive slots for master-to-slave transmission and immediate slave-to-master response. For SCO links the master periodically polls the corresponding slave, instead the polling is asynchronous for ACL links.

The SCO link can be considered as a circuit-switched connection between the master and the slave. The second kind of physical link, the ACL link, is a connection between the master and all slaves participating to the piconet, and it can be considered as a packet-switched connection between the Bluetooth devices that supports point-to-multipoint transmissions from the master to the slaves. The ACL channel guarantees the reliable delivery of data: a fast Automatic Repeat Request (*ARQ*) scheme is adopted to assure data integrity.

The MAC layer also accomplishes the Segmentation and Reassembly (*SAR*) procedure to improve the protocol efficiency by supporting a maximum transmission unit size (*MTU*) larger that the ACL packet sizes.

Table 1. ACL packets' maximum payload sizes (bytes)

DH_1	DH_3	DH_5	DM_1	DM_3	DM_5
27	183	339	17	121	224

It is worth pointing out that a tight constraint of Bluetooth technology is that ACL packets can covers 1, 3 or 5 time slots. According to Bluetooth specification [1], the ACL packets are of two different groups, one denoted DMx and the other one denoted DHx. The former has a payload encoded with a rate 2/3 FEC and the latter hasn't got a FEC encoding. The subscript x stands for the number of slots that are necessary to transmit the packet. Table 1 reports the different payload size of ACL packets.

More details about the Bluetooth Mac protocol can be found in [1], [3].

3. Efficient Double Cycle Scheduling Algorithm for a Bluetooth Piconet

The Efficient Double-Cycle (*EDC*) algorithm is based upon two main ideas: first of all it is necessary to avoid NULL transmissions towards and from the slaves; furthermore the fairness typical of a Round Robin scheme should be preserved. These targets can be accomplished if the selection of the slave to be polled takes into consideration the master knowledge of traffic from and to the slaves. In the rest of our discussion, we assume a system model where the master employs a separate queue for each slaves participating to piconet communications. It is worth pointing out that the master has a deterministic knowledge of the state of its local queues, i.e., the queues containing the packets for the slaves. On the other direction (i.e., traffic from the slaves), we only assume that the master has a probabilistic knowledge based on the feedback the master gets when polling the slaves. Therefore the master can only estimate the probability that a slave will send a NULL packet by exploiting the knowledge of the slave behavior in the previous polling cycles. As stated in the introduction, our goal is to define an efficient scheduling algorithm that is not implementation dependent, namely, it does not require that all slaves are able to set specific header fields according to the particular scheduling algorithm adopted by the master. Hereafter, we indicate as *uplink* the link direction from slaves to master, and as *downlink* the link direction from master towards slaves.

An additional problem to have a fair and efficient scheduling in Bluetooth is caused to the coupling between the transmission in uplink and downlink (i.e. a master to slave transmission implies also a polling of the slave and hence a, possibly NULL, slave to master transmission). Therefore, it is not possible to remove a slave from the polling cycle without blocking at the same time the master transmissions towards this slave. To introduce a (partial) decoupling in the scheduling of the transmissions in uplink and downlink we introduce the idea of a double polling cycle: an *uplink polling sub cycle*, hereafter called $Cycle_{UP}$, and a *downlink polling sub cycle*, hereafter called $Cycle_{DW}$. During both of these cycles the master selects a subset of

slaves that are *eligible* for the polling. For the sake of brevity we will refer to the subset of eligible slaves selected during a $Cycle_{UP}$ as *E(UP)*, and to the subset of eligible slaves selected during a $Cycle_{DW}$ as *E(DW)*. *E(DW)* is calculated considering only the state of master local queues, i.e. the ongoing traffic loads, whereas *E(UP)* is calculated considering only the estimated slaves' activity, i.e. the ingoing traffic loads. The distinction between polling rules for the downlink and the uplink permits to have a "*fairness separation*": in the *downlink* (*uplink*) *sub cycle* fairness is guaranteed only in the downlink (uplink) direction.

3.1 EDC specification

A detailed EDC specification through pseudo-code can be found in [10]. Due to the space constraints we provide only an EDC specification through the natural language.

The outcome of our polling algorithm is the next slave to be polled, also referred as *next*. Because there is a double cycle, we have a *nextUp* calculated during the $Cycle_{UP}$, and a *nextDw* calculated during the $Cycle_{DW}$. A polling interval c_i and a polling window w_i are associated to each slave S_i. These variables are used to estimate the slave activity, and to implement a *truncated binary exponential backoff* mechanism to control the slaves polling frequency during a $Cycle_{UP}$.. Specifically, at the end of each S_k's packet reception, the master updates the polling interval c_k and the polling window w_k with the following rules:

1. If S_k sends a packet with a null payload, then its polling window w_k is doubled till a maximum value w_{MAX} is reached.
2. If S_k sends a packet with a not null payload, then w_k is settled to be 1.
3. The polling interval c_k takes the w_k value.

During a $Cycle_{DW}$ all the S_i that belongs to *E(DW)* are polled in a cyclic order (i.e. we adopt a Round Robin policy for the *E(DW)* set). After a slave has been polled it is extracted from *E(DW)*. Therefore a $Cycle_{DW}$ is completed only when *E(DW)* becomes empty. At the same way, during a $Cycle_{UP}$ all the S_i that belongs to *E(UP)* are polled in a cyclic order (i.e., we adopt a Round Robin policy for the *E(UP)* set). After a slave has been polled it is extracted from *E(UP)*. Therefore a $Cycle_{UP}$ is completed only when *E(UP)* becomes empty. We consider a polling cycle as completed if all the slaves that belong to both *E(UP)* and *E(DW)* have been polled.

One of the most important tasks that is accomplished by the master at the beginning of each polling sub cycle is the update of *E(UP)* or *E(DW)* according to its knowledge about traffic loads. Specifically, *E(UP)* is constituted by all slaves S_i that have c_i null at the beginning of a $Cycle_{UP}$. Because *E(UP)* is updated before the beginning of each $Cycle_{UP}$, if S_i has a c_i equal to n, then n $Cycle_{UP}$ must elapse before S_i is polled during a $Cycle_{UP}$. On the other hand, *E(DW)* is constituted by all slaves S_i to which the master has traffic to send at the beginning of a $Cycle_{DW}$, i.e., slaves corresponding to the not-empty master local queues.

At the beginning of each new master transmission, according to the piconet timing, the master executes the following actions:

1. If the polling cycle is finished then EDC updates *E(DW)* and a new polling cycle begins with a $Cycle_{DW}$.
2. If the polling cycle is not finished, EDC determines if the next transmission belongs to a $Cycle_{DW}$, i.e., *E(DW)* is not an empty set, otherwise EDC decreases the polling interval variables, it updates *E(UP)* and a new $Cycle_{UP}$ begins.

It is worth noting that if at the beginning of a new polling cycle *E(DW)* is an empty set then EDC decreases the polling interval variables and a new $Cycle_{UP}$ begins. This behavior corresponds to have a $Cycle_{DW}$ with null duration. If *E(UP)* and *E(DW)* are both empty sets EDC has no information to discriminate a slave from the others, then EDC applies a round robin polling rule till the master does not receive a data packet from a slave or at least one of its local queues is not empty.

In the following section we present a complete simulative analysis of the EDC algorithm performances when it is used to schedule Internet traffic. By exploiting this analysis we demonstrate the performance enhancement achieved by EDC, with respect to a round robin scheduling algorithm. Furthermore, we point out some specific issues related to the transport of TCP traffic over the Bluetooth link to give a better understanding of EDC behavior.

4. Simulation Model and Performance Evaluation

The network model simulated is a single piconet constituted by a master and up to seven slaves. A detailed description of the Bluetooth architecture and its protocol stack can be found in [1], [2]. For the sake of the following discussion we refer to Figure 1, where we report a simplified architecture of the system.

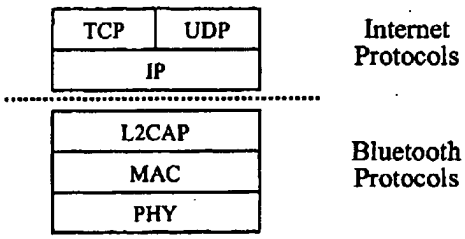


Figure 1. Protocol stack

The traffic sources generate TCP and UDP packets that are sent to the network layer, which adds its header (20 bytes). The L2CAP layer receives the data segments from the upper layers, it adds 4 bytes of L2CAP header and it queues the L2CAP packets in a buffer. The size of this buffer is chosen equal to 64 Kbytes that is the standard *advertised window* used in the TCP receiver. Since no more that an advertised window of TCP traffic can be generated by a TCP sender, this buffer size guarantees that no buffer overflow losses are experimented by TCP traffic[1] It means that each TCP source see an infinite size buffer. This assumption permits to study the dynamic of TCP connections under an EDC scheduler without biasing effects due to the limited buffer size.

Large L2CAP packets must be segmented into smaller Baseband packets before their transmission can take place. A new L2CAP packet cannot be served till all fragments (generated during the segmentation at the MAC layer) of the previous L2CAP packet have been successfully transmitted. The segmentation procedure is accomplished, just before the transmission, in such a way to maximize the amount of data conveyed by each baseband packet. In particular, the segmentation procedure executes the following steps:

1. Divide the L2CAP packet into an integer number of 5 slot baseband packets.
2. If the size x remaining to be fragmented is larger than the size of a 3 slot baseband packet, sent it as a 5 slot packet.

[1] Let us remind that a slave establishes a single TCP connections with the master

3. If x is larger than the size of a 1 slot baseband packet but shorter than the size of a 3 slot baseband packet, send it as a 3 slot baseband packet.
4. Otherwise, send x as a 1 slot baseband packet.

It would be possible to devise several SAR procedures (see [8], [9]), but this issue is left for future discussion.

Figure 2 shows the network configuration and traffic parameters, hereafter referred to as *scenario A*, used in the experiments presented in the next section. The TCP sources are asymptotic connections, i.e., they have always a data packet ready to be transmitted (ftp-like traffic). The TCP version considered is the TCP-Reno, the most worldwide adopted TCP implementation [5]. The UDP sources generate CBR traffic. In the figure the arrows indicate the direction of data packets in the connection slave-master. For each connection, the type of traffic (i.e., UDP or TCP), the time instant of beginning ad termination, and the bit rate (only for the UDP sources) are given.

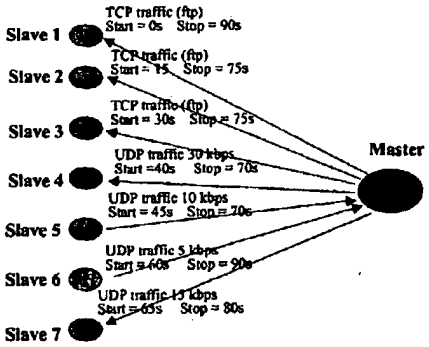


Figure 2. Traffic sources used in Scenario A

We consider sources with different time interval of activity to capture the dynamic behavior of the scheduling algorithm. We also locate some sources in the master and other sources in the slaves to evaluate the fairness of scheduling algorithm between the uplink and downlink direction. Finally, we analyze the ability of the scheduling algorithm to adapt to asymmetric loads by considering different CBR rates.

During all the simulations carried out we have assumed w_{MAX} equal to 32, and we have used ACL packets with no FEC encoding, i.e., only DHx packets (see Table 1).

4.1 Numerical comparison between EDC and RR behavior

In this section we consider and ideal channel with no errors, and we use a constant TCP packet size of 1024 bytes, a constant UDP packet size of 500 bytes and a TCP ACK size of 20 bytes.

Figure 3 shows the throughput for the TCP connection of Slave1 achieved in the scenario A with a master that adopts either the EDC algorithm or the Round Robin (RR) algorithm. We observe that EDC guarantees a throughput always significantly higher than the one achieved with a RR scheduler.

In the time interval [0, ... 15sec] there is a single active TCP connection. During this time interval the throughput obtained with EDC is more than two times the one obtained with RR, due to the capacity of EDC to adapt the polling frequency to the sources' activity. The ripples observed after 40 seconds are due to the starting of the UDP sources. The same results have been obtained for the other TCP connections and they are not reported here.

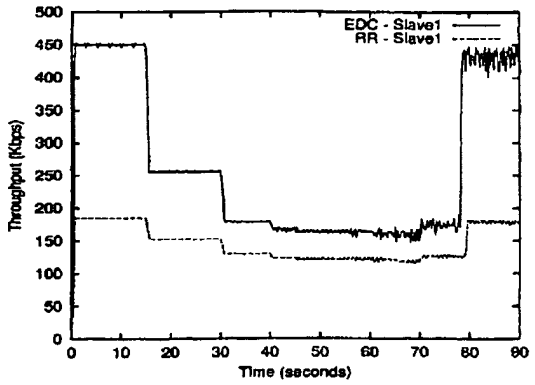


Figure 3. TCP throughput of Slave1 connection

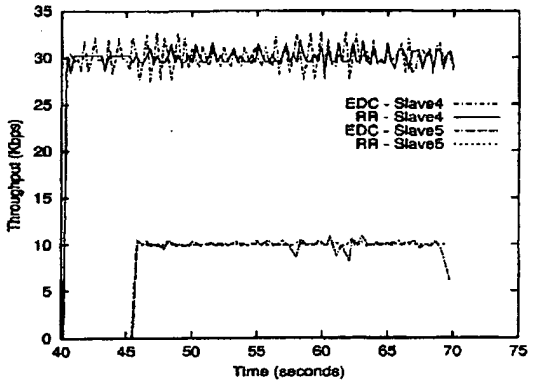


Figure 4. UDP throughput of Slave4 and Slave5

Figure 4 shows the throughput of two CBR sources, the former located in the master (curve labeled as Slave4) and the latter located in a slave (curve labeled as Slave5). It is worth noting that in the link slave4–master all the UDP packets are scheduled during the *downlink polling sub cycle*, whereas in the link slave5–master all the UDP packets are scheduled during the *uplink polling sub cycle*. However, the scheduler gives a fair treatment to these flows. Furthermore, we observe that EDC increases the throughput of TCP flows with respect to RR, but it also permits to guarantee a throughput for UDP flows equal to their rate, as RR.

Figure 5 shows the throughput of the three TCP connections when they are contemporarily active. We observe that EDC behaves fairly with the TCP flows, as a RR scheduler does. We have proved the fairness of EDC when all the TCP data packets are scheduled during $Cycle_{DW}$. In the following experiment we check if the different polling rules adopted during $Cycle_{DW}$ and $Cycle_{UP}$ can have an impact over the TCP performances.

Figure 6 shows the throughput of a TCP connection established between the slave2 and the master when the TCP sender is either located in the master (scenario A), or in the slave (scenario B). All the other sources behave exactly as in scenario A. This experiment shows that the TCP throughput slightly increases when the data packets flow is from the slave towards the master. This is due to the different frequency by which the scheduler polls slave2 during $Cycle_{UP}$ in the two cases.

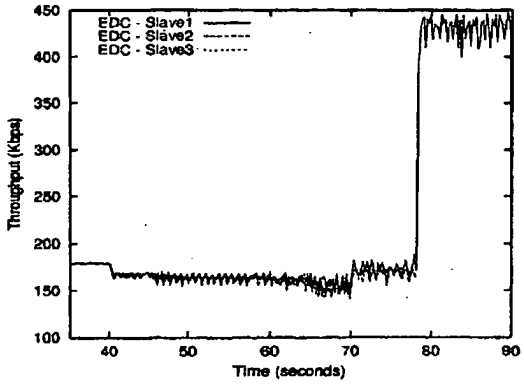


Figure 5. Fairness of EDC algorithm with respect to TCP connection

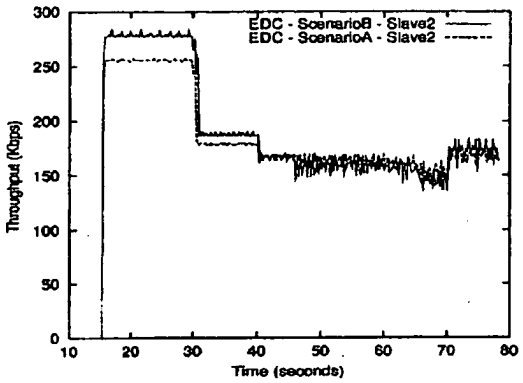


Figure 6. TCP throughput of Slave2 connection when reversing the connection direction

In scenario A, the slave2 queue contains the acknowledgement traffic. When the master sends a fragment of a TCP packet, it is highly probable that it receives a NULL packet from the slave2 (the ACK cannot be generated by the TCP receiver till the TCP packet is completely received), therefore the polling interval for slave2 increases, and the slave2 will skip some of the following *uplink polling sub cycles*. This behavior can reduce the rate by which the acknowledgement traffic is served, therefore also lowering the rate of TCP data (let us remind that the polling of one direction also means the polling of the other one). On the other hand, in scenario B the slave2 queue contains the data traffic, and it is highly probable to find a queued TCP packet (the TCP source is asymptotic), therefore no $Cycle_{UP}$ is skipped. Furthermore, as soon as the ACK for the slave2 is generated, the master will serve it in the first available $Cycle_{DW}$, with no additional delay. Obviously, this phenomenon is more evident when a few sources are active, because the polling cycle is short.

In conclusion, the results presented so far demonstrate that EDC significantly improves the throughput performance of TCP flows in a piconet, when compared to a RR scheduler. Furthermore, we have clarified that the decoupling of scheduler decisions related to the polling of the uplink and downlink can imply some issues when the traffic flows from a slave to the master and in the inverse direction are correlated (as for a TCP connection).

4.2 Effect of Maximum Segment Size over the TCP throughput

In this section we study in depth the role that the *Maximum Segment Size (MSS)* of a TCP connection has in determining the TCP throughput. In the following we assume that the TCP packet has a constant size equal to the MSS.

Figure 7 shows the throughput achieved from a TCP connection established from a master to a single slave versus the MSS. Obviously, when the packet size increase, the overhead due to the IP header and the L2CAP header has a decreasing weight. For example, with a MSS equal to 315, 654, 993 or 1332 bytes the delivery of a single TCP packet requires (including all upper layer overheads) exactly an integer number of DH_5 packets (1, 2, 3 or 4, respectively). Therefore the throughput increase is due to the decrease in the percentage of the overheads due to the upper layer. However, the dominant factor in determining the total throughput is the segmentation accomplished by the MAC layer. Specifically, as stated in Section 2, the segmentation mechanism has to use packets with discrete and fixed lengths.

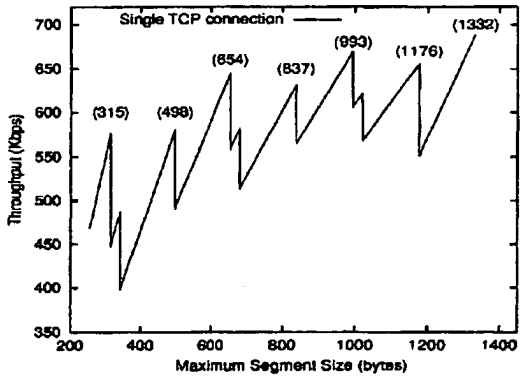


Figure 7. TCP throughput vs. MSS

For example, with a MSS=993 bytes, the delivery of a single TCP packet requires (including all upper layer overheads) exactly three DH_5 packets. If we increase the MSS to 994 bytes, then we need a further DH_5 packet, which conveys only a single byte of information. Therefore we measure a sharp decrease in the throughput, as reported in Figure 7. By exploiting similar considerations, we can easily understand the behavior of the TCP throughput curve shown in Figure 7.

These results presented so far, it follows that the selection of the MSS value for a TCP connection is a critical task in the Bluetooth technology. In the next experiment, we investigate the behavior of EDC and RR when the TCP connections adopt different MSSs. Specifically, we have considered a piconet with a master that establishes seven TCP connections towards as many slaves. Figure 8 shows the throughput obtained by each connection. The MSS adopted by each slave is reported over the related column. We have performed this experiment with both EDC and RR scheduler. We observe that EDC does not fairly assign the bandwidth to the flows. In particular the connection with the smallest MSS takes the highest throughput. This is due to the different ACK generation rate that characterizes the TCP connections.

Due to the delayed ACK mechanism, an ACK is generated at least after the reception of two TCP packets (after the reception of a single TCP packet if the time interval between two consecutive receptions is greater than 200 msec) [5]. The slave1 generates a number of ACK greater than

the other connections, since each ACL packet sent by the master towards the slave1 delivers a new TCP packet. Therefore slave1's queue has more traffic to send than the other slaves' queues, and the EDC polls more frequently slave1 that the other slaves during a $Cycle_{UP}$. With RR all the slave receive the same polling frequency, therefore the throughput difference between the connections are at most related to the different percentage of the ACL packets occupied by the upper layer overheads (see Figure 7). It is worth pointing out that the aggregate throughput achieved with EDC is greater than the one achieved with RR, in particular 624 Kbytes against 611 Kbytes, even if all the TCP source are asymptotic.

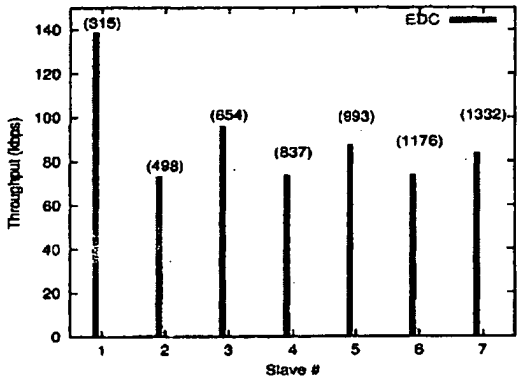


Figure 8. TCP throughput of connections with different MSSs

Figure 8 seems to suggest that RR behaves better than EDC for some MSSs, like MSS=1332. More precisely, from Figure 8 we can only derive that if a TCP connection adopts a short MSS can become more aggressive towards TCP connections with a long MSS, from a throughput standpoint.

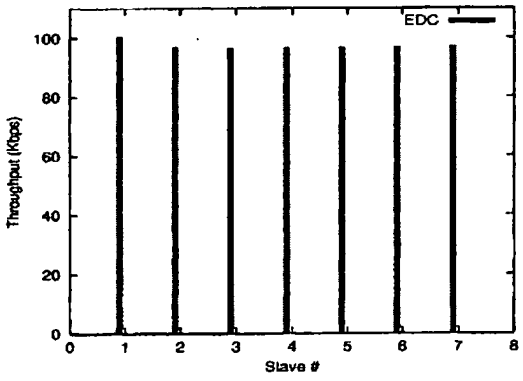


Figure 9. TCP throughput of connections with MSS=1332

Figure 9 shows the throughput obtained by each connection when all of them adopt a MSS=1332. We observe that EDC is fair as RR when all the TCP connection have the same MSS. Furthermore EDC guarantees a greater throughput than RR even if all connections are asymptotic. It is worth noting that the performance gain guaranteed by EDC increase with not-asymptotic TCP flows. Specifically, the EDC behavior is the same of RR behavior when all the queues are never empty. In the previous section, we have shown that EDC is able to adapt its polling frequency to the traffic characteristics when a source alternates periods of activity with period of inactivity. Hence, the analysis with asymptotic TCP flows is a worst-case analysis. The same results are obtained with the other MSSs.

In conclusion, we can assess that TCP connections with a small MSS are more aggressive than the ones with a large MSS when the EDC algorithm is adopted. Further studies are required to introduce a fair behavior of the scheduler also towards TCP connections with different MSSs.

4.3 Effect of the ARQ scheme adopted in Bluetooth over TCP behavior

In this section we evaluate the impact of channel errors on the TCP behavior. In particular, we are interested in investigating the effectiveness of the unnumbered ARQ scheme[2] adopted by the Bluetooth MAC layer to hide the channel errors to the transport layer.

We model the wireless channel as a discrete two-state Markov Chain [11]. In each state the bit error rate (BER) is constant, but in one state, i.e. the *Bad state*, the BER is significantly higher than in the other one, i.e. the *Good state*. Hereafter, we will consider the BER in Good state equal to 2x10E-6, and the BER in Bad state equal to 10E-4. We assume that, in the Markov chain model, the average sojourn time in each state is expressed as a multiple T of the time slot. Furthermore, we assume that the BER remains constant during the packet transmission. We have also considered a different model for the wireless channel with uniformly distributed errors. In this case the BER has been chosen equal to the average bit error rate experimented in the two-state model. This model is used to investigate the impact of error recovery mechanism over the TCP both with burst of errors and uniformly distributed errors.

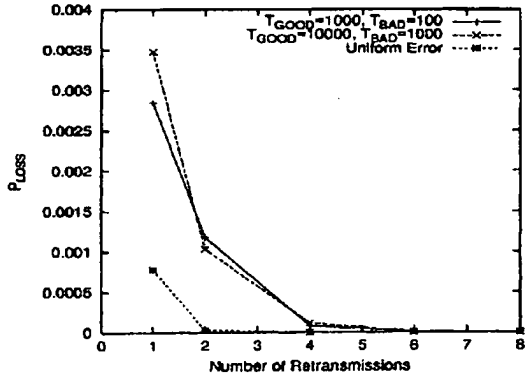


Figure 10. TCP packet loss probability with MSS=1332 bytes

It is worth pointing out that the main objective of this section is to illustrate the behavior of the ARQ scheme in presence of bursty losses on the wireless channel, hence an approximate characterization of the wireless channel is sufficient. To the best of our knowledge, an exhaustive characterization of the piconet wireless environment based on experimental measures is not yet available, and the numerical values chosen in this paper are compliant with the one adopted by other researchers [9], [12].

We have studied the impact of the maximum number of consecutive retransmissions for the same fragment, say it

[2] A Stop and Wait like error-recovery mechanism [1]

ritr_thresh, on the TCP packet-loss probability. It is worth pointing out that when a fragment is discarded because the *ritr_thresh* is exceeded, the MAC layer discards all the subsequent fragments belonging to the same TCP packet. It is possible because the MAC layer itself executes the segmentation procedure.

Figure 10 shows the TCP packet loss probability for a connection that adopts a MSS=1332 bytes versus the *ritr_thresh*. We observe that the unnumbered ARQ scheme utilized by the MAC layer is very efficient, and for a *ritr_thresh* equal to 4 the lost TCP packets are less then one every 10000 packets sent. Furthermore, we observe that a bursty channel is more critical than a uniform channel, because the concentrated errors cause to close the congestion window for the TCP more rapidly than spare errors. Finally, in the two bursty cases considered there are not significant differences.

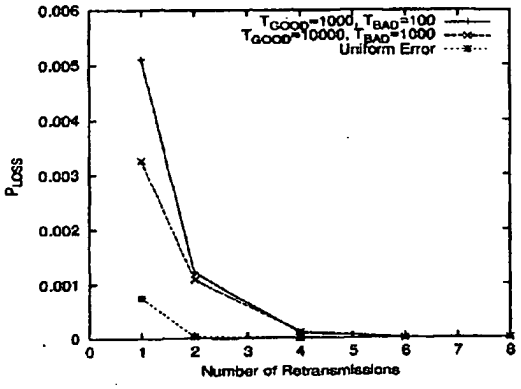


Figure 11. TCP packet loss probability with MSS=315 bytes

Figure 11 shows the same experiments when the connection adopts a MSS=315. In Figure 11, the TCP packet-loss probability shows the same behavior as in figure 10. Hence, similar considerations can be taken.

We can conclude that the ARQ scheme adopted by Bluetooth is effective to hide the channel errors to the transport layer, just with a *ritr_thresh* of 4.

5. Conclusions and Future Work

The paper proposes a new scheduling algorithm, named EDC, for the Bluetooth MAC layer. EDC is evaluated when the Bluetooth technology is used as the wireless technology for the Internet access. EDC exploits master knowledge of local queues' occupancy to avoid NULL packet transmission, and it employs two polling cycles with different polling rules to guarantee a separate and fair treatment of uplink and downlink connections. EDC significantly improves the throughput of TCP connections when compared to a RR scheduler.

In this paper we have investigated the importance of the MSS parameter in determining the TCP throughput. In particular, we have shown that the presence of TCP connections that adopt different MSSs causes unfairness. Further studies are necessary to improve the fairness of Bluetooth technology with regard to TCP connections with different characteristics. In all the experiments executed in this work we have only considered sources placed in either the master or a slave. However the traffic sources can be located in any point in the Internet. The impact of the delays introduced when the TCP flows cross the real Internet network will be studied in future works.

6. References

[1] Bluetooth Special Interest Group, "Specification of the Bluetooth System 1.0b, Volume 1: Core", December 1999.

[2] B. A. Miller, C. Bisdikian, "Bluetooth Revealed", Prentice Hall, 2000.

[3] R. Bruno, M. Conti, E. Gregori. Chapter 4: "Traffic Integration in Personal, Local and Geographical Wireless Networks", in *"Handbook of Wireless Networks and Mobile Computing"*, John Wiley & Sons, New York.

[4] R. Bruno, M. Conti, E. Gregori, "WLAN technologies for mobile ad hoc networks", Proc. *Hawaii International Conference on System Sciences, (HICSS-34)*, Maui, Hawaii, January 3-6, 2001.

[5] W. Richard Stevens, "TCP/IP Illustrated, Volume 1: The Protocols", Addison-Wesly , 1994.

[6] N. Johansson, U. Korner, P. Johansson, "Performance Evaluation of Scheduling Algorithm for Bluetooth", Proc. *IFIP Broadband Communications*, Hong Kong, November 1999.

[7] N. Johansson, M. Kihl, U. Korner, "TCP/IP over Bluetooth Wireless Ad-hoc Network", Proc. IFIP Networking 2000, Paris, May 1999, pp. 799-810.

[8] M. Kalia. Deepack Bansal, R. Shorey, "Data Scheduling and SAR for Bluetooth MAC", Proc. *IEEE VTC 2000*, Tokyo, Japan. May 2000.

[9] A. Das, A. Ghose, A. Razdan, H. Saran, R. Shorey, "Efficient Performance of Asynchronous Data Traffic over Bleutooth Wireless Ad-hoc Network", Proc. *IEEE INFOCOM 2001*, Anchorage, AK, USA, April 2001.

[10] R. Bruno, M. Conti, E. Gregori, "Bluetooth: architecture, protocols and scheduling algorithms". *Cluster Computing*, to be published.

[11] M. Zorzi, R.R. Rao, "On the statistics of block errors in bursty channels", *IEEE Trans on Comm.*, Vol. 45, N.6 1997, pp. 660-667.

[12] S. Zurbes, W. Stahl, K. Matheus, J. Haartsen, "Radio Network Performance of Bluetooth", Proc. *ICC 2000*, New Orleans, USA, June 2000, pp.1563-1567.

Wireless access to internet via Bluetooth: performance evaluation of the EDC scheduling algorithm

Full text Pdf (568 KB)

Source **Wireless Mobile Internet** archive
Proceedings of the first workshop on Wireless mobile internet table of contents
Rome, Italy
Pages: 43 - 49
Year of Publication: 2001 JULY
ISBN:1-58113-423-1

Authors Raffaele Bruno Consiglio Nazionale delle Ricerche, Istituto CNUCE, Via G. Moruzzi, 1, 56124 Pisa, Italy
Marco Conti Consiglio Nazionale delle Ricerche, Istituto CNUCE, Via G. Moruzzi, 1, 56124 Pisa, Italy
Enrico Gregori Consiglio Nazionale delle Ricerche, Istituto CNUCE, Via G. Moruzzi, 1, 56124 Pisa, Italy

Sponsor SIGMOBILE: ACM Special Interest Group on Mobility of Systems, Users, Data and Computing

Publisher ACM New York, NY, USA

↑ ABSTRACT

Bluetooth is an emerging technology for constructing adhoc wireless Personal Area Networks (WPANs). In this paper we analyze an innovative scheduling algorithm for asynchronous data traffic specifically tailored to the Bluetooth characteristic. This algorithm, named Efficient Double-Cycle (EDC), dynamically adapts the polling frequency to the traffic conditions. By considering a scenario where a Bluetooth master is used as wireless access point to the Internet, we show that our EDC scheduler significantly enhances the system performance with regard to a Round Robin (RR) scheduler.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 Bluetooth Special Interest Group, "Specification of the Bluetooth System 1.0b, Volume 1: Core", December 1999.

2 Brent A. Miller , Chatschik Bisdikian , Tom Siep, Bluetooth Revealed, Prentice Hall PTR, Upper Saddle River, NJ, 2001

3 Raffaele Bruno , Marco Conti , Enrico Gregori, Traffic integration in personal, local, and geographical wireless networks, Handbook of wireless networks and mobile computing, John Wiley & Sons, Inc., New York, NY, 2002

4 R. Bruno , M. Conti , E. Gregori, WLAN Technologies for Mobile ad hoc Networks, Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 9, p.9003, January 03-06, 2001

5 W. Richard Stevens, TCP/IP illustrated (vol. 1): the protocols, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1993

6 Niklas Johansson , Ulf Körner , Per Johansson, Performance evaluation of scheduling algorithms for Bluetooth, Broadband communications: convergence of network technologies, Kluwer Academic Publishers, Norwell, MA, 2000

7 Niklas Johansson , Maria Kihl , Ulf Körner, TCP/IP over the Bluetooth Wireless Ad-hoc Network, Proceedings of the IFIP-TC6 / European Commission International Conference on Broadband Communications, High Performance Networking, and Performance of Communication Networks, p.799-810, May 14-19, 2000

8 M. Kalia. Deepack Bansal, 1L Shorey, "Data Scheduling and SAR for Bluetooth MAC", Proc. /EEE PTC 2000, Tokyo, Japan. May 2000.

9 A. Das, A. Ghose, A. Razdan, H. Saran, R. Shorey, "Efficient Performance of Asynchronous Data Traffic over Bleutooth Wireless Ad-hoc Network", Proe. IEEE INFOCOM2001, Anchorage, AK, USA, April 2001.

10 Raffaele Bruno , Marco Conti , Enrico Gregori, Bluetooth: Architecture, Protocols and Scheduling Algorithms, Cluster Computing, v.5 n.2, p.117-131, April 2002 [doi>10.1023/A:1013989524865]

11 M. Zorzi, ILR. Rao, "On the statistics of block errors in bursty channels", IEEE Trans on Comm., Vol. 45, N.6 1997, pp. 660-667.

12 S. Zurbes, W. Stahl, K. Mathens, J. Haartsen, "Radio Network Performance of Bluetooth", Proc. ICC 2000, New Orleans, USA, June 2000, pp.1563-1567.

↑ CITED BY *4*

Daniele Miorandi , Andrea Zanella , Gianfranco Pierobon, Performance evaluation of Bluetooth polling schemes: an analytical approach, Mobile Networks and Applications, v.9 n.1, p.63-72, Feburary 2004

Gwo-Jen Hwang , Judy C. R. Tseng , Yu-San Huang, I-WAP: An Intelligent WAP Site Management System, IEEE Transactions on Mobile Computing, v.1 n.2, p.82-95, April 2002

Marco Conti, Body, personal, and local ad hoc wireless networks, The handbook of ad hoc wireless networks, CRC Press, Inc., Boca Raton, FL, 2003

Rachid Ait Yaiz , Geert Heijenk, Polling Best Effort Traffic in Bluetooth, Wireless Personal Communications: An International Journal, v.23 n.1, p.195-206, October 2002

↑ INDEX TERMS

Primary Classification:

C. Computer Systems Organization

↳ **C.2** COMPUTER-COMMUNICATION NETWORKS

↳ **C.2.1** Network Architecture and Design

↳ **Subjects:** Wireless communication

Additional Classification:

C. Computer Systems Organization

↳ **C.2** COMPUTER-COMMUNICATION NETWORKS

F. Theory of Computation

↳ **F.2** ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY

↳ **F.2.2** Nonnumerical Algorithms and Problems

↳ **Subjects:** Sequencing and scheduling

General Terms:
Algorithms, Performance

Keywords:
Medium Access Control (MAC), TCP, automatic repeat request (ARQ), bluetooth, polling, scheduling

↑ **Collaborative Colleagues:**

Raffaele Bruno: Stefano Basagni, Marco Conti, Enrico Gregori, Gabriele Mambrini, Chiara Petrioli

Enrico Gregori: Gi useppe Anastasi, Stefano Basagni, Luciano Bononi, Eleonora Borgia, Raffaele Bruno, Federico Calì, Frederico Calì, Andrew T. Campbell, Ludmila Cherkasova, Marco Conti, Gianpaolo Cugola, Franca Delmastro, Silvia Ghezzi, Willy Lapenna, Luciano Lenzini, Silvia Martelli, Cambyse Guy Omidyar, Fabio Panzieri, Andrea Passarella, Gian Pietro Picco, Ioannis Stravrakakis, Giovanni Turi, Moshe Zukerman

↑ **Peer to Peer - Readers of this Article have also read:**

- Data structures for quadtree approximation and compression **Communications of the ACM** 28, 9
  Hanan Samet
- A hierarchical single-key-lock access control using the Chinese remainder theorem **Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing**
  Kim S. Lee , Huizhu Lu , D. D. Fisher
- The GemStone object database management system **Communications of the ACM** 34, 10
  Paul Butterworth , Allen Otis , Jacob Stein
- Putting innovation to work: adoption strategies for multimedia communication systems **Communications of the ACM** 34, 12
  Ellen Francik , Susan Ehrlich Rudman , Donna Cooper , Stephen Levine

- An intelligent component database for behavioral synthesis **Proceedings of the 27th ACM/IEEE conference on Design automation**
  Gwo-Dong Chen , Daniel D. Gajski

Multimedia Support Over Bluetooth Piconets

Rohit Kapoor
UCLA
3803 H, Boelter Hall,
UCLA, Los Angeles
USA - 90024
rohitk@cs.ucla.edu

Manthos Kazantzidis
UCLA
3803 D, Boelter Hall
UCLA, Los Angeles
USA - 90024
kazantz@cs.ucla.edu

Mario Gerla
UCLA
3732 F, Boelter Hall
UCLA, Los Angeles
USA - 90024
gerla@cs.ucla.edu

Per Johansson
Ericsson
3803 B, Boelter Hall
UCLA, Los Angeles
USA - 90024
per.johansson@ericsson.com

ABSTRACT

In this paper we explore the ability to support multimedia traffic in indoor, wireless ad hoc PANs (Personal Area Networks) using the Bluetooth technology. We first define the representative ad hoc networking applications such as wireless access to the Internet, document distribution, videoconferencing, webcasting, interaction with sensors and actuators, etc. For such applications, we define the performance requirements placed on the PAN. There are two technologies now competing for the PAN market: the IEEE802.11 "legacy" technology, and the newly introduced Bluetooth technology. By IEEE802.11, we refer to the operation of 802.11 in the DCF mode, which is the mode implemented in the commercial WaveLAN cards. In the rest of the paper, we will use WaveLAN to refer to 802.11 in its DCF mode. We will attempt to answer the questions: how effective is the Bluetooth technology in supporting collaborative, "virtual ad hoc networking" applications and how does it compare with WaveLAN? To answer these questions, we have developed an NS-2 model of Bluetooth. We have also developed models of adaptive applications such as voice and video. For WaveLAN, we have used the existing NS-2 models. The results show that Bluetooth provides better support for real-time applications as compared to WaveLAN. It does not exhibit the "capture" behavior observed, for example, in WaveLAN. Also, with the addition of nodes to the "indoor" space, it adds to the total "system" capacity and gives a better overall throughput.

Categories and Subject Descriptors

QoS support in wireless access networks

General Terms

Measurement, Performance, Experimentation

Keywords

Bluetooth, Multimedia, Piconets, Video, Voice, WaveLAN.

1st Workshop on Wireless Mobile Internet 7/01 Rome, Italy
© 2001 ACM ISBN 1-58113-423-1/01/07...$5.00

1 INTRODUCTION - Ad hoc networking and Personal Area Networks (PANs)

With the increasing dependence on the Internet in many aspects of their daily lives, users demand ubiquitous, high performance Internet access whether they are at work, at home, or on the move. Moreover, users on the move are often interested in forming "ad hoc" networks to collaborate with colleagues at conferences, or more generally to interconnect all their personal devices. This type of network, which is centered on the individual himself, is often called the **Personal Area Network** (PAN).

The PAN is defined as the collection of devices carried by a mobile, networked individual (e.g., a professional on the move, an Internet-wise tourist, a student attending "virtual classes", an avid Internet game player, etc). The devices include any subset of: cell phone, laptop, earphones, GPS navigator, palm pilot, beeper, portable scanner, etc. These devices form his/her PAN (also known as personal "bubble"). The connectivity within the bubble is wireless (using for example a low cost, low transmit power wireless LAN such as Bluetooth). The bubble can expand and contract dynamically depending on needs. The bubble may connect to wall repeaters for **access to the Internet**. It may also be dynamically stretched to include **access to sensors and actuators**. Such access is critical when the "nomad" walks into a new environment and wants to quickly become aware of what is going on, or wants to control temperature, adjust the lighting, select a particular background music etc. In some cases, the nomad himself carries sensors as part of his PAN: for example, a patient may walk around in the hospital or nursing home with several monitors which transmit to repeaters on the walls, allowing customized 24 hour monitoring.

The PAN communication infrastructure should enable efficient support of the above ad hoc networking scenarios. In essence, we need a self-configuring communications infrastructure that can: (a) provide efficient multimedia access from the PAN to the Internet, (b) permit communications with various classes of sensor/actuators, and (c) enable voice/data intra and inter-PAN networking. The key challenges in the design of the PAN protocol architecture are: (a) the design of middleware and adaptive application protocols that provide smooth transition between different bandwidth, connectivity and mobility configurations, and; (b) the implementation of PAN MAC and network layer protocols and their interconnection with existing public (wired and wireless) network infrastructures.

In the **single PAN environment**, where nodes are all within transmission range of each other, key issues are (1) MAC protocol selection, to provide efficient transport of TCP/IP traffic and at

JULY 2001

the same time satisfy multimedia traffic requirements; (2) efficient handoff; (3) mobile/cellular IP support; and (4) end to end adaptivity, possibly via proxy agents. When communicating with sensors, the PAN MAC and network layer protocols must operate in a connectionless, low latency and low overhead mode. In this paper, we focus on the "single PAN environment" operation of the PAN's, where communication occurs only within a PAN, and evaluate the support of multimedia in such an environment.

2 THE SCOPE OF THIS STUDY

In this study, we will assume that each PAN corresponds to a single user and consists of a portable device (e.g., laptop, PDA, etc.). We will limit ourselves to an application of the PAN where all communication occurs within the PAN.

In this simplified, single hop setting the performance of the network will be for the most part determined by the MAC layer. Currently, there are two leading candidates for such a role: (a) the IEEE 802.11 MAC protocol and (b) the Bluetooth MAC protocol [1]. The IEEE 802.11 protocol is a rather sophisticated protocol that includes a fairly broad range of options. In particular, it includes the PCF (Point Coordination Function) mode which permits a "base station" to poll various terminal in a cellular-type environment. It also includes the DCF (Distributed Coordination Function) mode, which supports peer-to-peer, ad hoc type communications. The DCF version is a random access protocol similar to CSMA, with the addition of RTS and CTS (for collision avoidance) and of an ACK returned by the receiver after successful transmission. In our study we will assume the use of the DCF mode, which is the mode implemented in the WaveLAN cards.

A couple of years ago a new MAC protocol was proposed as part of the Bluetooth PAN architecture. The Bluetooth MAC protocol is a major departure from the IEEE802.11 protocol. To start with, it uses Frequency Hopping instead of Direct Sequence Spread Spectrum, thus exhibiting better protection from co-channel interference. Secondly, it uses time/slot synchronization. Moreover, it uses a polling type scheme to allow a "master" to poll the "slaves" in a given cluster. Bluetooth is expected to become very popular due to its low cost (in the order of a few dollars per interface). The details of the Bluetooth protocol are provided in the next section. Here, it suffices to say that the enormous commercial interest in these two PAN candidates and at the same time their markedly different characteristics warrants an in depth comparison of their performance in various realistic indoor scenarios.

In our simulation experiments we have recreated scenarios that are typical of indoor ad hoc networking. We will consider a mixed traffic environment, both with data (TCP) and with voice/video streaming (with fixed and adaptive rate).

We will be interested in the throughput and delay measures, and in the fairness behavior exhibited by the two schemes. The simulation results will be reported in Sec 4. In the next section we first introduce the Bluetooth architecture and protocols.

3 BLUETOOTH OVERVIEW

The Bluetooth system operates in the worldwide unlicensed 2.4 GHz Industrial-Scientific-Medical (ISM) frequency band. To make the link robust to interference, it employs a Frequency Hopping (FH) technique, in which the carrier frequency is changed at every packet transmission. To minimize complexity and to reduce the cost of the transceiver, a simple binary Gaussian frequency shift keying modulation is adopted. In order to allow efficient wideband data transmission the bit rate is 1 Mbit/s.

Two or more Bluetooth units sharing the same channel form a piconet, see Fig. 1(a).

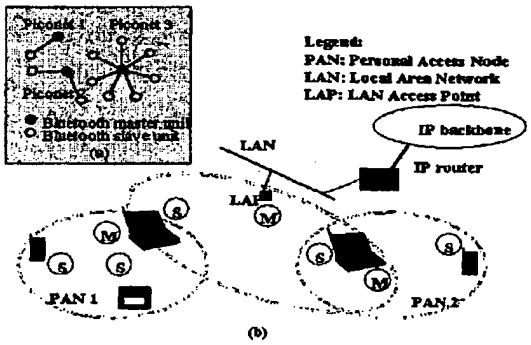


Figure 1: (a) Bluetooth Piconet (b) Bluetooth Scatternet

Within a piconet a Bluetooth unit can be either master or slave. Within each piconet there may be only one master (and there must always be one) and up to seven active slaves. Any Bluetooth unit can become a master in a piconet. Furthermore, two or more piconets can be interconnected, forming what is called a scatternet, see Fig. 1(b). The connection point between two piconets consists of a Bluetooth unit that is a member of both piconets. A Bluetooth unit can simultaneously be a slave member of multiple piconets, but a master in only one, and can only transmit and receive data in one piconet at a time, so participation in multiple piconets has to be on a time division multiplex basis.

The Bluetooth system provides full-duplex transmission using a slotted time division duplex (TDD) scheme where each slot is 0.625 ms long. Master-to-slave transmissions always start in an even-numbered time slot, while slave-to-master transmissions always start in an odd-numbered time slot. An even-numbered time slot and its subsequent odd-numbered time slot together are called a frame. There is no direct transmission between slaves in a Bluetooth piconet; transmission is only between a master and a slave, and vice versa.

The communication within a piconet is organized such that the master polls each slave. A slave is only allowed to transmit after the master has polled it. The slave will then start its transmission in the slave-to-master time slot immediately following the packet received from the master.

Each Bluetooth unit has a globally unique 48-bit IEEE 802 address. This address is permanently assigned when the unit is manufactured. In addition to this, the master of a piconet assigns a local active member address (AM ADDR) to each active member of the piconet. The AM ADDR is three bits long, is dynamically assigned and reassigned, and is unique only within a single piconet. The master uses the AM ADDR when polling a slave in a piconet.

Bluetooth packets can carry either synchronous data on synchronous connection oriented (SCO) links mainly intended for

voice traffic, or asynchronous data on asynchronous connectionless (ACL) links. To ensure reliable transfer of data, a fast acknowledgment and retransmission scheme is used, only for ACL links. In addition, a forward error correction (FEC) scheme may be used to further improve reliable packet transmission.

4 CASE STUDIES AND SIMULATION RESULTS

In this Section, we present simulation results based on a set of representative traffic scenarios. One of the main goals was to evaluate achievable Bluetooth throughput taking into account interference between difference coexisting piconets.

4.1 Simulation Environment

The simulation environment used in our experiments is NS-2 [2]. NS-2 already includes several wireless network models. In particular, it supports the IEEE 802.11 WaveLAN standard. We have augmented NS-2 with the Bluetooth model. The Bluetooth model has support for defining multiple piconets that may overlap with each other causing interference. The model contains most of the standard features of Bluetooth like Frequency Hopping, Multi-Slot Packets, Fast ARQ (Automatic Retransmission Query).

An important feature of the simulator is the Channel Model. Frequency hopping is modeled as a pure pseudo-random sequence. If two or more transmissions occur on the same frequency, the SIR (Signal-to-Interference Ratio) is evaluated using the gain factor g of each radio channel. The factor g is considered constant during the packet transmission and its value is obtained by considering the following factors:

- *path loss due to distance*: $d^{-\eta}$, where d is the distance and η ranges between 2 and 4.
- *shadowing*: log-normal random variable $s=10^{0.1\varepsilon}$, where ε is a Gaussian variable with σ standard deviation.
- *fading*: exponential random variable (Rayleigh fading) with mean = 1.

At the receiver i, the SIR is evaluated as:

$$SIR = \frac{P_i \cdot g_{ii}}{P_n + \sum P_j \cdot g_{ji}}$$

where g_{ij} is the gain factor between transmitter j and receiver i, P_t is the transmitted power and P_n is the noise power in the signal bandwidth.

The receiver model is based on the SIR value. For each portion of the packet of length L in which the SIR is considered constant the information bit error probability is evaluated by taking into consideration the modulation adopted and the FEC coding (if adopted). The probability that the portion of the packet is correct is calculated assuming independent errors in the segment.

The Bluetooth slave polling strategy that we have used is the one given in [3]. It tries to assign slots to slaves based on their traffic history and activity. The topology is totally static, which means that nodes are not mobile and piconets are set up at the beginning of the simulation and do not dynamically change. Again, it is important to note that connections are only 1 or 2-hops, as in intra-piconet communication. No inter-piconet communication takes place.

4.2 Conference Hall Case Study

Our aim is to compare the performance of Bluetooth and WaveLAN in a totally ad-hoc environment, where no infrastructure in the form of access points is available. This would typically model the scenario of a large conference, where a number of Bluetooth or WaveLAN devices may be talking to each other. The traffic in such a scenario is heterogeneous and multimedia in nature, i.e., TCP, voice and video. It is assumed here that any two devices wanting to communicate are close enough to be in the same piconet and thus communicate through the master. This will be a realistic model for ad-hoc group collaboration where members of the same team will be sitting nearby and will interact with each other by exchanging files and engaging in videoconference.

In the experiment, we consider a 50m * 100m room, in which nodes are distributed according to a uniform random distribution. In the case of Bluetooth, piconets are formed by clustering the nodes close enough to each other. The number of slaves present in each piconet is chosen randomly. Also, some piconets overlap with each other incurring a certain fraction of collisions. The traffic consists of a mix of TCP, Voice and Video. The number of TCP, voice and video connections are in the ratio 1:1:1. Unless otherwise stated, the connections start at 0.5s and run till the end of the simulation. The TCP data connections are always active large file backlogs, with 500-byte packets. The voice connections are modeled according to the Brady model [4]. In particular, the voice connections are "on-off" sources. The on and off times are exponentially distributed, with mean 1 s and 1.35 s respectively. The voice-coding rate is 8 kbit/s and the packetisation period is 20 ms, which gives a payload size of 20 bytes. Header compression is assumed for voice packets in Bluetooth and the total packet size is 30 bytes. Voice packets are sent using RTP over UDP. Each experiment lasts 32 seconds of simulation time. In order to probe the sensitivity of performance to population size and to the number of simultaneous connections, we perform different experiments choosing different values of number of nodes and connections.

4.3 Video Traffic

The video sources are represented by real traces. We use the Star Wars trailer clip encoded using Intel's H.263 compatible codec. The traces have been smoothed using a simple technique, namely a frame as returned by the codec is distributed uniformly in time within the frame interval using a target of 200 byte packets. There is no other smaller time scale transport mechanism and the generated packets are simply sent through the network with UDP. A few seconds from the resulting sources for the codec is shown in Fig 2. A description of the framework used in the experiments can be found in [5]. The platform used in the experiments is based on ns, augmented to support video agents and end-to-end adaptive video agents [7].

We investigate adaptive as well as non-adaptive video streaming. The former uses average rates of 48, 64, 80, 128 and 256Kbps for the two codecs, while the non-adaptive cases use the 256Kbps trace. The adaptation mechanism is based on an end-to-end, periodic (1 sec) feedback that contains the number of packets received during the feedback interval. This feedback is used by the server to compute the RTP loss rates. The server then changes its rate using a min/max loss threshold.

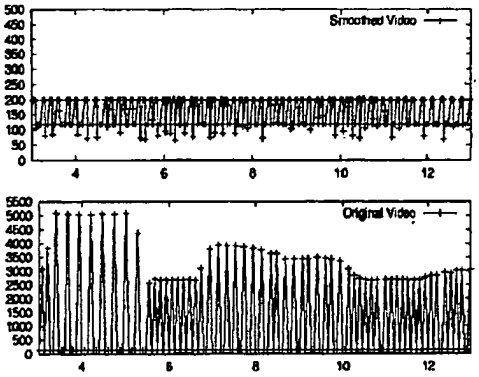


Figure 2:A few seconds from the H263 source trace (sec, bytes)

Below the minimum packet loss rate (5%) the server attempts to additively increase its rate. When the loss rate is above the maximum loss threshold (15%) the server reduces the sending rate, choosing a rate among the 5 available rates that is appropriate to support the reported loss rate. For example, if the current rate is 128Kbps and the loss rate is 50% the sending rate drops to 64Kbps.

The following graphs (Fig 3(a) and (b)) illustrate this behavior in a random 30-node 60-connection experiment. This initial experiment targets at showing the adaptive behavior with the two MAC protocols. The scenario is generated as mentioned in Section 4.2, but now 90% of the voice, video and TCP connections are created at 8.6s and finish at the 16.6s. The goal is to study the adaptive behavior of a video connection that lives throughout the experiment (i.e. from 0s to 32s). As the feedback indicates, the server downgrades the sending rate or attempts a higher rate. We show the loss rates as calculated when a feedback packet is received, the per-packet delays and the server selected average rates for the two cases. First, we note that when the additional connections enter the network (from 8.6s to 16.6s) in WaveLAN, the video connection downgrades to the lower possible transmission rate because the loss feedback goes beyond the threshold. On the other hand, in Bluetooth the loss rates are lower, the transmission rates remain higher and the downgrading is in all cases gradual (one layer at a time). These indicate that the network response is more regular allowing for efficient feedback control with fewer oscillations. This is true not only for the connection shown but for the other competing adaptive connections as well. It is interesting to note that in the congested network, fewer packets get lost in the Bluetooth case, but their delay is significantly increased. This can be explained by the difference in the Bluetooth and WaveLAN MAC layers. WaveLAN retransmits a collided packet a finite number of times and then drops the packet if transmission is unsuccessful. Since collisions are very high for large number of connections in WaveLAN, a larger number of packets get dropped. In Bluetooth, on the other hand, the chances of a collision are very low because of the controlled access to the channel and the frequency hopping behavior. Thus, a larger number of packets reach the destination, but these fill up the link layer queues and make delays larger.

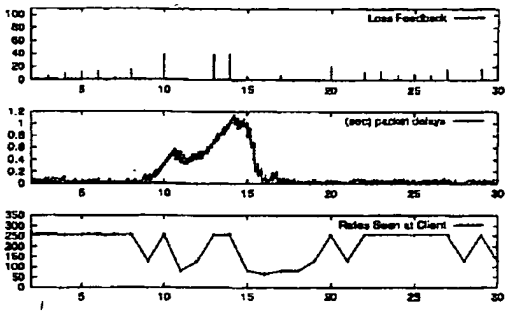


Figure 3(a): Bluetooth End-to-End Adaptation

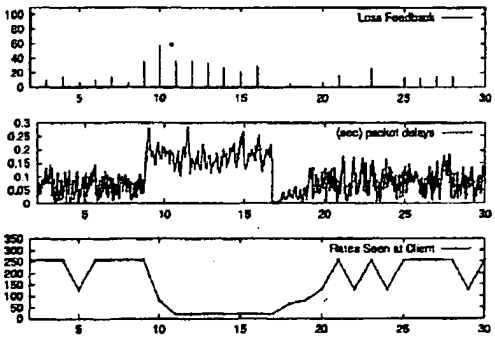


Figure 3(b): WaveLAN End-to-End Adaptation

4.4 Aggregate results –Non-Adaptive Video, TCP and Voice

We first performed the experiments described in Section 4.2 with TCP, Voice and Non-Adaptive Video connections.

4.4.1 Video and TCP

The aggregate throughput is the same for all experiments for WaveLAN. Bluetooth, however, manages to grow the aggregate throughput as the number of nodes increases. The smaller range and formation of additional piconets adds capacity to the network.

A generic difference in the way the two source types, TCP and Video, share the network bandwidth is illustrated in Fig 4. In WaveLAN, individual TCP connections are allowed to grow their window and 'capture' the channel. When this happens, video connections suffer increased loss rates. On the other hand, the presence of polling in Bluetooth allows the video connections to share the channel with the TCP. In fact, with Bluetooth, the video achieves its full rate for different configurations. Several measurements are reported in Fig. 4 as shown by the caption below each sample (Number of nodes; number of connections; WL vs. BT). It can also be noted from Fig. 4 that the total throughput for WaveLAN is higher than that of Bluetooth for the 30 nodes, 30 connections case. As the number of connections increases, the total throughput for Bluetooth increases with respect to WaveLAN since larger connections means more collisions in WaveLAN. Also, a larger number of nodes causes the total throughput for Bluetooth to be higher since it leads to

formation of more piconets and hence, addition to system capacity.

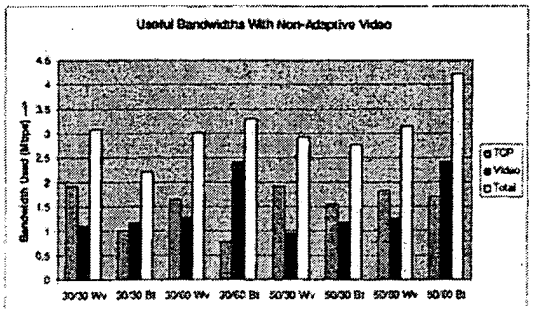


Figure 4: H.263 Non adaptive video and TCP connections aggregate throughput.

From Fig. 5, we see that the loss rates for video are much higher in the WaveLAN case where the contention between the connections, especially TCP and video, allows some TCP connections to increase their window and capture the channel, locking out packets from the 256Kbps video connections. On the other hand Bluetooth shows less than 5% packet loss in all cases. Due to less packet retransmissions the Bluetooth case will save a significant amount of power that is particularly important in battery-powered devices.

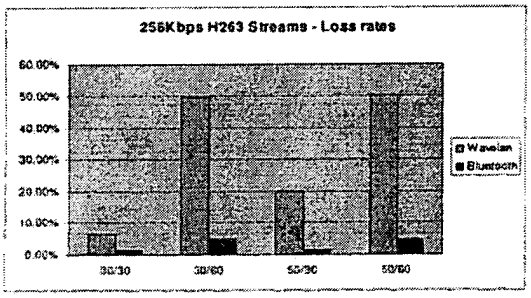


Figure 5: Loss Rates for video connections for H.263.

4.4.2 Voice

The significant parameter that needs to be studied for voice is the delay. The complementary cumulative delay distributions for voice in Bluetooth and WaveLAN for 30 nodes and 60 connections with non-adaptive MPEG Video are shown in Figs. 6(a) and (b).

It is seen that the delays suffered are much lower in Bluetooth than in WaveLAN. From the complementary cumulative distribution graph, we note that a packet loss ratio of less than 5% can be obtained for a play-out buffer of about 80 ms in the case of Bluetooth, whereas a play-out buffer of more than 350 ms is required to achieve the same effect with WaveLAN. Typically, a delay in excess of 300ms is considered unsuitable for interactive voice communications.

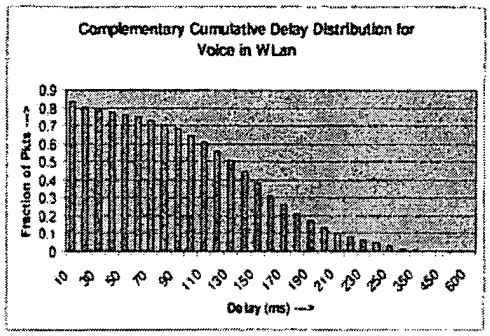


Figure 6 (a): Voice Delay Distribution for WaveLAN

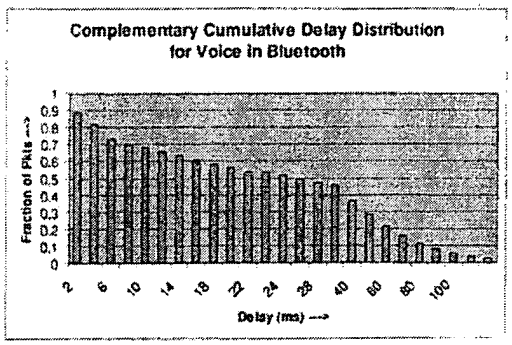


Figure 6 (b): Voice Delay Distribution for Bluetooth

To explain the very high WaveLAN delays, recall that the scenario considered here is of a very congested network with large number of connections. In such a network, the uncontrolled access to the channel and the large number of collisions and retransmissions in case of WaveLAN leads to large delays. Bluetooth, on the other hand, has a very controlled access to the channel determined by the polling scheme. This keeps the delays low and well bounded.

4.5 Aggregate results with Adaptive Video

In this section we repeat the experiments of Section 4.2 with adaptive in place of non-adaptive video. The video sources adapt through the use of a periodic end-to-end feedback containing the RTP loss rates, as described in Section 4.3.

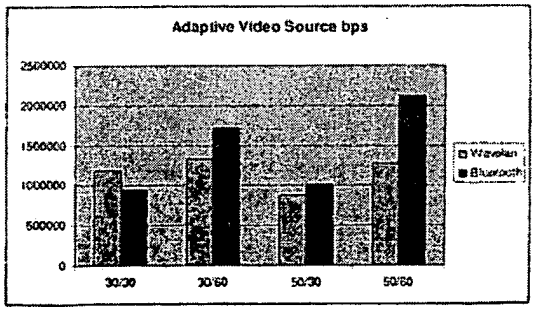


Figure 7: H.263 aggregate server sent rates

First, we show the aggregate source sending rates in Fig 7. This quantity is influenced by the loss rates reported to the server and represents the application requested bandwidth. The sum of the feedback packets received in the WaveLAN configurations was 1215 whereas in Bluetooth 1216 feedback packets were received. WaveLAN tends to transmit less, especially at high connection density and load, showing that a smaller number of packets were received at the destination.

Next we look at the loss rates with adaptive video in Fig 8. The controlled, adaptive polling environment of Bluetooth, with less reverse channel problems managed to eliminate the video loss rates almost completely in the adaptive case in our experiments.

The highest aggregate loss rate in Bluetooth is 1.32%. In WaveLAN too the loss rates are reduced to half with respect to the non-adaptive case shown in Fig 5.

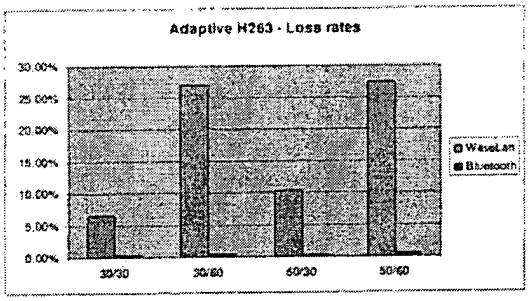


Figure 8: Loss Rates for Adaptive H.263 Video

Next we examine adaptive video throughput in Fig 9. 1Mbps Bluetooth throughputs are again comparable to the 10Mbps WaveLAN total throughput. As video connections adapt they allow TCP connections to get more bandwidth. In Bluetooth video adaptation reduces loss rate to less than 1% in most cases whereas in WaveLAN adaptive video connections suffer 25% to 30% loss rates.

The total throughput is higher in WaveLAN than in Bluetooth for lower number of nodes. As more piconets are formed, Bluetooth adds bandwidth and surpasses the constant WaveLAN bandwidth, which is independent of the number of nodes.

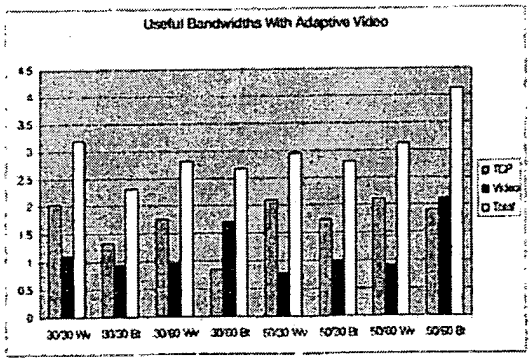


Figure 9: H.263 adaptive video and TCP connections aggregate throughput.

5. CONCLUSIONS

In this paper we have evaluated the efficacy of the Bluetooth technology in supporting ad hoc indoor communications. The simulation results show that Bluetooth performs very well in mixed data and real time traffic scenarios typical of such environments. In particular, it guarantees service quality to multimedia streams while providing fair share of capacity to TCP users. It does not suffer from the TCP capture behaviour exhibited by WaveLAN. Though the total system throughput is larger for WaveLAN for small number of nodes, Bluetooth can exceed the WaveLAN throughput when number of nodes becomes large, by using multiple, overlaid piconets. Adaptive video applications fare better with Bluetooth than WaveLAN, in part because the polling schedule of Bluetooth seems to offer a more stable service to adaptive video, precluding oscillations. It is to be noted again that these experiments were performed with the DCF mode of 802.11. In the future, we plan to repeat some of the experiments using the PCF mode.

Work is currently in progress in several directions. A Scatternet model is being developed, to allow the interconnection of piconets. Sensor interaction experiments are planned, with various mobility models.

6. REFERENCES

[1] J. Haartsen, BLUETOOTH - the universal radio interface for ad hoc wireless connectivity, Ericsson Review, n.3, 1998, pp. 110-117.

[2] Network Simulator (NS-2), www-mash.cs.berkeley.edu/ns/

[3] A. Capone, R. Kapoor and M. Gerla: Efficient Polling Schemes for Bluetooth Picocells, ICC 2001.

[4] P.T. Brady: A model for generating on-off speech patterns in two-way conversation, Bell System Technical Journal, Sept. 1969, pp. 2445-2471.

[5] M.I Kazantzidis, I. Slain, Y. Romanenko, T. Chen, M. Gerla: Experiments on QoS adaptation for improving end user speech perception over multi-hop wireless networks, IEEE International Conference on Communications IEEE, 1999. pp. 708-15

[6] M. Gerla, P. Johansson, R. Kapoor, and F. Vatalaro: Bluetooth: last meter technology for nomadic wireless internetting, 12th Tyrhennian International Workshop on Digital Communications.

[7] M.I. Kazantzidis, L. Wang, M. Gerla: On fairness and efficiency of adaptive audio application layers for multihop wireless networks, 1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99) IEEE, 1999. p.357-62. xii+390 pp.

Multimedia support over bluetooth Piconets

Full text Pdf (808 KB)

Source **Wireless Mobile Internet** archive
Proceedings of the first workshop on Wireless mobile internet table of contents
Rome, Italy
Pages: 50 - 55
Year of Publication: 2001 July
ISBN:1-58113-423-1

Authors
Rohit Kapoor UCLA, 3803 H, Boelter Hall, UCLA, Los Angeles
Manthos Kazantzidis UCLA, 3803 D, Boelter Hall, UCLA, Los Angeles
Mario Gerla UCLA, 3732 F, Boelter Hall, UCLA, Los Angeles
Per Johansson Ericsson, 3803 B, Boelter Hall, UCLA, Los Angeles

Sponsor SIGMOBILE: ACM Special Interest Group on Mobility of Systems, Users, Data and Computing

Publisher ACM New York, NY, USA

↑ ABSTRACT

In this paper we explore the ability to support multimedia traffic in indoor, wireless ad hoc PANs (Personal Area Networks) using the Bluetooth technology. We first define the representative ad hoc networking applications such as wireless access to the Internet, document distribution, videoconferencing, webcasting, interaction with sensors and actuators, etc. For such applications, we define the performance requirements placed on the PAN. There are two technologies now competing for the PAN market: the IEEE802.11 "legacy" technology, and the newly introduced Bluetooth technology. By IEEE802.11, we refer to the operation of 802.11 in the DCF mode, which is the mode implemented in the commercial WaveLAN cards. In the rest of the paper, we will use WaveLAN to refer to 802.11 in its DCF mode. We will attempt to answer the questions: how effective is the Bluetooth technology in supporting collaborative, "virtual ad hoc networking" applications and how does it compare with WaveLAN? To answer these questions, we have developed an NS-2 model of Bluetooth. We have also developed models of adaptive applications such as voice and video. For WaveLAN, we have used the existing NS-2 models. The results show that Bluetooth provides better support for real-time applications as compared to WaveLAN. It does not exhibit the "capture" behavior observed, for example, in WaveLAN. Also, with the addition of nodes to the "indoor" space, it adds to the total "system" capacity and gives a better overall throughput.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 J. Haartsen, BLUETOOTH - the universal radio interface for ad hoe wireless connectivity, Ericsson Review, n.3, 1998, pp. 110-117.

2 Network Simulator (NS-2), wwwmash.cs.berkeley.edu/ns/

3 A. Capone, R. Kapoor and M. Gerla: Efficient Polling Schemes for Bluetooth Picocells, ICC 2001.

4 P.T. Brady: A model for generating on-off speech patterns in two-way conversation, Bell System Technical Journal, Sept. 1969, pp. 2445-2471.

5 M.I Kazantzidis, I. Slain, Y. Romanenko, T. Chen, M. Gerla: Experiments on QoS adaptation for improving end user speech perception over multi-hop wireless networks, IEEE International Conference on Communications IEEE, 1999. pp. 708-15

6 M. Gerla, P. Johansson, R. Kapoor, and F. Vatalaro: Bluetooth: last meter technology for nomadic wireless internetting, 12 th Tyrhennian International Workshop on Digital Communications.

7 M.I. Kazantzidis, L. Wang, M. Gerla: On fairness and efficiency of adaptive audio application layers for multihop wireless networks, 1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99) IEEE, 1999. p.357-62, xii+390 pp.

↑ CITED BY

Myra Dideles, Bluetooth: a technical overview, Crossroads, v.9 n.4, p.11-18, June 2003

↑ INDEX TERMS

Primary Classification:

C. Computer Systems Organization

↳ **C.2** COMPUTER-COMMUNICATION NETWORKS

↳ **C.2.1** Network Architecture and Design

↳ **Subjects:** Wireless communication

Additional Classification:

C. Computer Systems Organization

↳ **C.2** COMPUTER-COMMUNICATION NETWORKS

H. Information Systems

↳ **H.5** INFORMATION INTERFACES AND PRESENTATION (I.7)

↳ **H.5.1** Multimedia Information Systems

↳ **Subjects:** Video (e.g., tape, disk, DVI)

General Terms:

Experimentation, Measurement, Performance

Keywords:

Piconets, WaveLAN, bluetooth, multimedia, video, voice

↑ Collaborative Colleagues:

Mario Gerla:

Nadeem Aboobaker
Khaldoun Al Agha
Hasan S. Alkhatib
Sang Bae
Sang Ho Bae
Rajive Bagrodia
Rajive L. Bagrodia
Chase Bailey
Kannan Balasubramanian
Alex Balk
Nicholas Bambos
Elizabeth Belding-Royer
Bharat Bhargava
G. Boiocchi
Flaminio Borgonovo
Khaled Boussetta
Mauro Brunato
Carlo Caini
Pietro Camarda
Claudio Casetti
Dirceu Cavendish
Dirceu Galvao Cavendish
David Chanady
Alexander Chang
Jiwei Chen
Ling-Jyh Chen
Ling-Jyh Chen
Shou C. Chen
Tsu-Wei Chen
Yu-an Chen
Guido Chiaretti
Fabio M. Chiussi
Renato Lo Cigno
Paola Crocetti
Babak Daneshrad
S. Das
Shirshanka Das
Siddhartha Yeshwant Devadhar
Joseph Evans
Maria Fazio
Aiguo Fei
Stefano Ferretti
Rosario Firrincieli
Michael Fitz
Shivi Fotedar
Luigi Fratta
Steve Friedman
Zhenghua Fu
Giorgio Gallassi
Daniel Lihui Gu
Xiaoyan Hong
Xiaoyan Hong
Xiaoyang Hong
San-Chiao Huang
Chia-Heng Hung
Vikas Jha
Zhengrong Ji
Per Johansson
Sewook Jung
Csaba Kiss Kallo
Rohit Kapoor
Manthos Kazantzidis
Matheos Ioannis Kazantzidis
Jinkyu Kim
Milan Kovačevic
Srikant Krishnamurthy
Bruce Kwan
Li Lao
Addison Lee
Scott Seongwook Lee
Sung Ju Lee
Sungwook Lee
Uichin Lee
Uichin Lee
Emilio Leonardi
Chun-Hung Lin
Chunhung Richard Lin
Ying-Dar Jason Lin
Jun Liu
Winston W. Liu
Pierpaolo Loreti
Songwu Lu
M. Luglio
Desmond S. Lun
Dario Maggiorini
Eugenio Magistretti
Cesar Marcondes
M. A. Marsiglia
Jay Martin
Saverio Mascolo
James McCrae
Bartlett S. H. Michel
Jelena Mirkovic
Prasant Mohapatra
Mart Molle
José Augusto Suruagy Monteiro
Jose Augusto Suruagy Monteiro
Alok Nandan
Fabio Neri
Bryan Kwok Fai Ng
Katia Obraczka
Soon Y. Oh
Fernando Paganini
Claudio E. Palazzi
Prasasth Palnati
Prasasth Reddy Palnati
Joon-Sang Park
Joon-Sang Park
Joon-Sang Park
Michael G. Parker
Giovanni Pau
Carlos M. D. Pazos
Carlos Marcelo Dias Pazos
Guangyu Pei
Gregorio Procissi
Lantao Qi
Ramesh Rao
Gianluca Reali
Peter Reiher
Ronn Allan Ritke
Marco Roccetti
Christian Roche
Raphael Rom
M. Y. Sanadidi
Yantai Shu
Edilayne M. Silva
Vijay Sivaraman
J. Stepanek
William Su
William Wei-Lien Su
Tony Sun
Tony Sun
Tony Sun
Yea-Li Sunny Sun
José Augusto Suruagy Monteiro
Tsung-Yuan Tai
Mineo Takai
Fabrizio Talucci
Edward Tsai
M. A. Vázquez-Castro
Massimo Valla
Francesco Vatalaro
Ren Wang
Dapeng Wu
Hsiao-Kuang Wu
Kaixin Xu
K. Yamada
Guang Yang
Guang Yang
Guang Yang
Joseph Yeh
Yunjung Yi
H. Yu
Andrea Zanella
Xiang Zeng
Petros Zerfos

Per Johansson:

Fredrik Alriksson
Mikael Degermark
Mario Gerla
Nicklas Hedman
Ulf Jönsson
Bartosz Mielczarek

	Ulf Körner	
	Rohit Kapoor	
	Manthos Kazantzidis	
	Tony Larsson	
	Gerald Q. Maguire	
Rohit Kapoor:	Ling-Jyh Chen	Francesco Vatalaro
	Mario Gerla	Andrea Zanella
	Per Johansson	
	Manthos Kazantzidis	
	Pierpaolo Loreti	
	M. Luglio	
	Alok Nandan	
	M. Y. Sanadidi	
	J. Stepanek	
	M. A. Vázquez-Castro	
Manthos Kazantzidis:	Mario Gerla	
	Per Johansson	
	Rohit Kapoor	
	Dario Maggiorini	
	Guangyu Pei	
	Fabrizio Talucci	